



UC3M – Grado en Ingeniería Electrónica y Automática Industrial

Optimización de la trayectoria del iCab basado en waypoints

Trabajo fin de Grado

Madrid, Junio 2016

Autor:

Miguel Prieto López

Directores:

Pablo Marín Plaza

Ahmed Ali Mahmoud Hussein

ÍNDICE GENERAL

Prefacio

Resumen	4
Abstract	4

1 Introducción y objetivos

1.1 Introducción	6
1.2 Objetivos	7

2 Estado del Arte

2.1 Modelo geométrico de Ackerman	9
2.1.1 Modelo del cuatriciclo	9
2.1.2 Modelo del biciclo	11
2.2 Métodos geométricos de “path tracking”	13
2.2.1 Follow the carrot	13
2.2.2 Pure Pursuit	15
2.2.3 Vector Pursuit	18
2.2.4 Comparación entre los métodos geométricos	24
2.3 Planteamiento del problema	25

3 Solución Propuesta

3.1 Obtención de la trayectoria a partir de los waypoints	28
3.1.1 Mínimos cuadrados	28
3.1.2 Ecuación de la recta	30
3.2 Implementación del algoritmo Pure Pursuit	34
3.2.1 Elección del parámetro “look ahead distance”	34
3.2.2 Cálculo del punto objetivo	36
3.2.3 Cálculo del ángulo de dirección	43
3.2.4 Actualización de la orientación	59
3.3 Diagrama de flujo del algoritmo completo	64

4 Resultados y Conclusiones

4.1 Resultados de las simulaciones	66
4.1.1 Criterio de selección de waypoints	68
4.1.2 Conclusiones sobre las simulaciones (número de waypoints)	72
4.1.3 Criterio para establecer el parámetro “look ahead distance”	73
4.1.4 Conclusiones sobre las simulaciones (look ahead distance)	76
4.2 Conclusiones y Líneas futuras	77

5 Planificación y presupuesto

5.1 Planificación	79
5.2 Presupuesto	80

Apéndice

Fuentes Bibliográficas	82
Anexo 1	83

ÍNDICE DE FIGURAS

Figura 1: Dimensiones del iCab [8]	6
Figura 2: Geometría Ackerman [1]	9
Figura 3: Trapecio de la dirección [1]	10
Figura 4: a) geometría Ackerman 4 ruedas; b) geometría Ackerman 2 ruedas [3].....	11
Figura 5: Modelo Ackerman, Bicicleta [2]	12
Figura 6: follow the carrot [4]	13
Figura 7: efectos en la elección del parámetro "look ahead distance" [5]	14
Figura 8: geometría "Pure Pursuit" [2].....	15
Figura 9: explicación parámetros "Pure Pursuit"	17
Figura 10: coordenadas Plücker, teoría de tornillos [6]	18
Figura 11: teoría de tornillos, movimiento del sólido rígido [6]	19
Figura 12: sistemas de referencia, vector pursuit [6]	20
Figura 13: circunferencias, vector pursuit [6]	22
Figura 14: cálculo del radio, Vector Pursuit [6].....	23
Figura 15: Ejemplo ruta de calculada	25
Figura 16: ejemplo de waypoints	26
Figura 17: ejemplo recta de aproximación	32
Figura 18: ejemplo generación de trayectoria, rectas consecutivas.....	33
Figura 19: radio mínimo de giro [8].....	35
Figura 20: obtención del punto objetivo.....	37
Figura 21: elección del punto objetivo, método 1	40
Figura 22: elección del punto objetivo, método 2	42
Figura 23: diferencia entre look ahead distance y distancia Goal	43
Figura 24: Parámetros Pure Pursuit	44
Figura 25: sistemas de referencia y vectores implicados en el cálculo del ángulo α	46
Figura 26: cálculo del centro de giro	48
Figura 27: elección del centro de giro correcto, método 1.....	51
Figura 28 : ángulos Beta1 y Beta2, método 2	55
Figura 29: criterio para el signo del ángulo de dirección método 2.	56
Figura 30: obtención del nuevo "heading"	60
Figura 31: cálculo del nuevo heading y criterio	62
Figura 32: ángulos Beta_1 y Beta_2	63
Figura 33: camino de ejemplo 1	66
Figura 34: camino de ejemplo 2	67
Figura 35: camino de ejemplo 3.....	67

Resumen

En el presente trabajo fin de grado se desarrolla un algoritmo de navegación para un vehículo autónomo. El algoritmo está basado en el método geométrico de “path tracking” Pure Pursuit. Los fundamentos teóricos del modelo Pure Pursuit, así como todos los problemas geométricos y otros detalles referentes a su implementación, se detallan en los capítulos 2 y 3 de este documento.

La técnica de navegación está basada en el uso de “waypoints”, es decir, el camino que debe seguir el vehículo está definido por una serie de puntos discretos en un plano XY. Normalmente los algoritmos de navegación no parten directamente de los waypoints, sino que consideran que ya existe una trayectoria dada.

El algoritmo desarrollado consta de dos partes diferenciadas. En primer lugar se crea la trayectoria a partir de los waypoints. Para lo cual es necesario aplicar algún método matemático de ajuste de curvas, es decir, encontrar una función “ $f(x)$ ” que se ajuste lo máximo posible a los “N” puntos seleccionados (waypoints).

En segundo lugar se aplica el modelo Pure Pursuit para calcular los comandos necesarios para la navegación autónoma del vehículo. Esto incluye el cálculo de un punto objetivo y el ángulo de dirección a aplicar para alcanzar dicho punto.

Abstract

This bachelor dissertation shows the development of a navigation algorithm for an autonomous ground vehicle. The algorithm is based on the Pure Pursuit path tracking method. All the theoretical principles and the relevant geometric issues are explained in detail on chapters 2 and 3.

The navigation technique used is based on waypoints. This means that the desired path, that the vehicle should follow, is defined by a distribution of points on a XY plane. Usually the navigation algorithms consider that there is already a given path to follow. However in this case the algorithm developed also creates a path using the waypoints.

Therefor the algorithm is divided in two parts. First of all, a path is created using a curve fitting method. Mathematically that means finding a function “ $f(x)$ ” which best fits the “N” current waypoints used.

Secondly, the Pure Pursuit method is used to calculate an achievable goal point on the defined path, and the necessary steering angle to command the vehicle.

Keywords: Pure Pursuit, autonomous vehicles, Path tracking techniques, waypoint, navigation algorithm, path follow.

/

Introducción y objetivos

1.1 Introducción

Este trabajo fin de grado forma parte de un proyecto mucho mayor que está siendo desarrollado por un grupo de investigadores del departamento de sistemas y automática de la universidad Carlos III de Madrid. El proyecto iCab abarca la conducción autónoma de vehículos terrestres AUGV (Autonomous Unmanned Ground Vehicle) [11].

El objetivo del proyecto, es lograr que la universidad disponga de una flota de vehículos autónomos, que circulen por el campus y sean capaces de llevar a los pasajeros a su destino sin intervención humana. Los vehículos en cuestión son unos carritos de golf como el de la Figura 1.

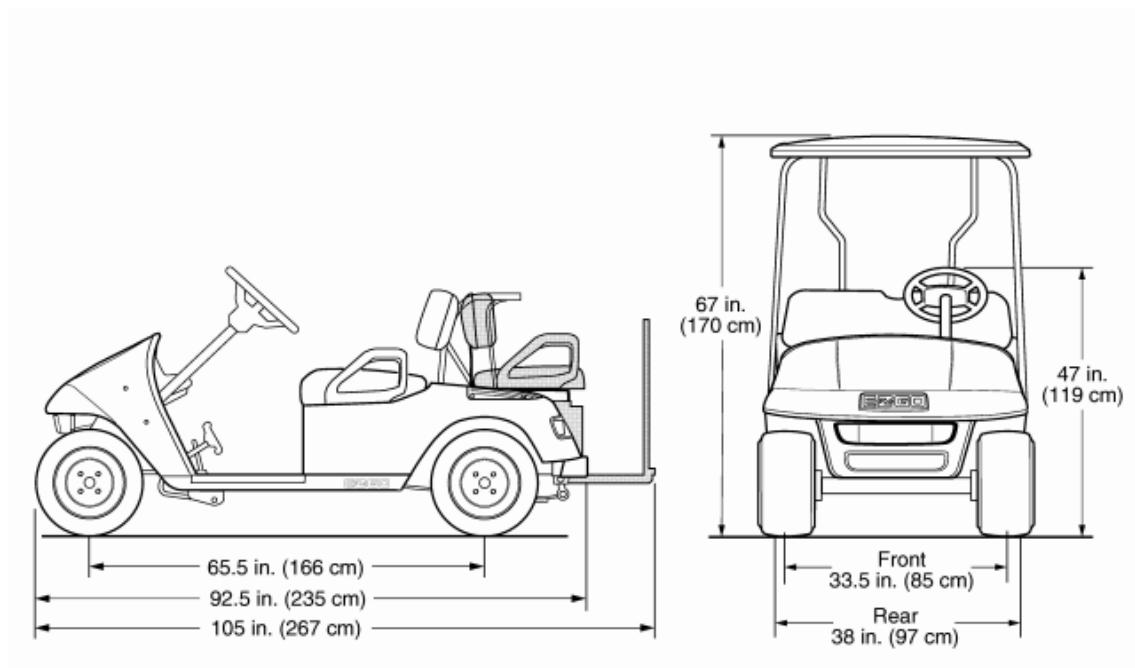


Figura 1: Dimensiones del iCab [8]

Actualmente existen dos prototipos con los que se está trabajando para desarrollar e implementar el sistema completo. Ambos vehículos han sido modificados mecánica y electrónicamente para hacer posible su autonomía.

Entre otros sensores, los vehículos están dotados de un sensor laser y cámaras stereo para la detección de obstáculos y la modelación del entorno. El control del vehículo se realiza mediante un ordenador a bordo, en el cual se ejecutan los algoritmos necesarios sobre el sistema operativo ROS.

1.2 Objetivos

Un sistema de navegación autónoma para vehículos terrestres abarca diversas tareas que deben complementarse unas a otras. Dichas tareas pueden clasificarse en las siguientes temáticas, en función de su objetivo:

- Mapeado y modelación del entorno
- Localización
- Planificación de trayectorias y navegación
- Detección y evasión de obstáculos

El objetivo de este trabajo fin de grado es desarrollar un algoritmo de navegación, que sea capaz de convertir un “camino de puntos” dado (waypoints) en instrucciones con las que el vehículo pueda navegar de manera autónoma.

Para que sea posible integrar el algoritmo de navegación en el sistema de control existente, se programará en lenguaje orientado a objetos, siendo C++ el idioma de programación elegido.

El sistema de control de los vehículos está basado en un PC sobre el que se ejecuta el sistema operativo ROS. Los distintos algoritmos de mapeado, localización, navegación etc. se ejecutan sobre dicho SO.

Aunque la finalidad del algoritmo es la implementación del código desarrollado en el vehículo y su posterior optimización., no forma parte de los objetivos del presente proyecto realizar la implementación real en el vehículo. Es decir, el código generado se limita a calcular los parámetros necesarios para la navegación del vehículo, y como componente adicional se simula el comportamiento teórico del vehículo, pero el algoritmo no se implementa sobre el sistema operativo ROS.

Al momento de empezar este trabajo fin de grado, el proyecto i-Cab ya está muy avanzado y existen múltiples algoritmos implementados en el sistema de control del vehículo. Razón por la cual trasladar el código desarrollado a ROS implicaría tener un conocimiento profundo de todo el trabajo ya existente.

Por consiguiente, este proyecto se limita a desarrollar un algoritmo de navegación en C++ que posteriormente pueda ser integrado en el sistema de control de los vehículos basado en ROS.

//

Estado del arte

2.1 Modelo geométrico de Ackerman

Dentro de esta sección se explica brevemente qué es el modelo de Ackerman, su utilidad y sus aplicaciones. Se analizan dos configuraciones, el cuadríciclo y el biciclo, ésta última como modelo simplificado de la primera.

2.1.1 Modelo del cuadríciclo

El modelo geométrico de Ackerman resuelve un problema subyacente en vehículos de cuatro ruedas con dirección delantera. El sistema de dirección de este tipo de vehículos suele estar basado en el principio de Ackerman.

Cuando un vehículo se dispone a trazar una curva, el ángulo de giro de la rueda delantera interior y exterior no es el mismo (Figura 2). Esto se hace para conseguir que las cuatro ruedas giren con respecto al mismo centro de rotación instantáneo, lo cual minimiza la aparición de fuerzas laterales no deseadas que producen un desgaste desigual de los neumáticos [1].

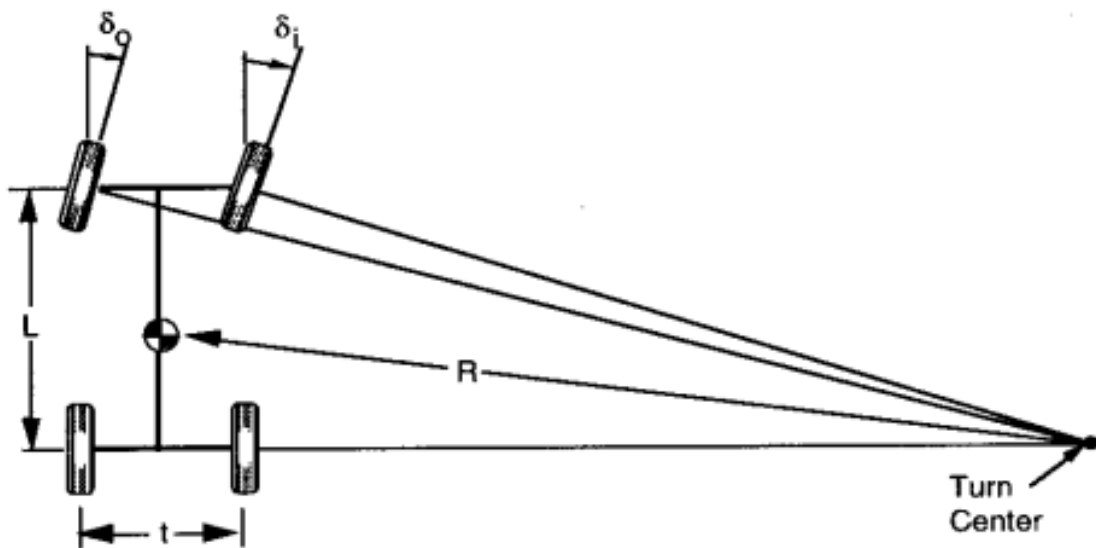


Figura 2: Geometría Ackerman [1]

Los ángulos δ_o y δ_i de la Figura 2 dependen de la distancia entre los ejes delantero y trasero, así como de la longitud de ambos ejes. δ_o y δ_i están definidos según las siguientes expresiones [1]:

$$\delta_o = \tan^{-1} \left(\frac{L}{R + t/2} \right) \quad [2.1]$$

$$\delta_i = \tan^{-1} \left(\frac{L}{R - t/2} \right) \quad [2.2]$$

En la práctica, los ángulos δ_o y δ_i no se calculan y aplican sobre la rueda correspondiente para cada caso, sino que se consideran condiciones de diseño para el sistema de dirección del vehículo. Como se ve en la Figura 3, una de las implementaciones más sencillas es el uso de un trapecio articulado, ya que debido a la asimetría que presenta dicha figura geométrica, se pueden conseguir ángulos distintos en sus vértices al aplicar un esfuerzo.

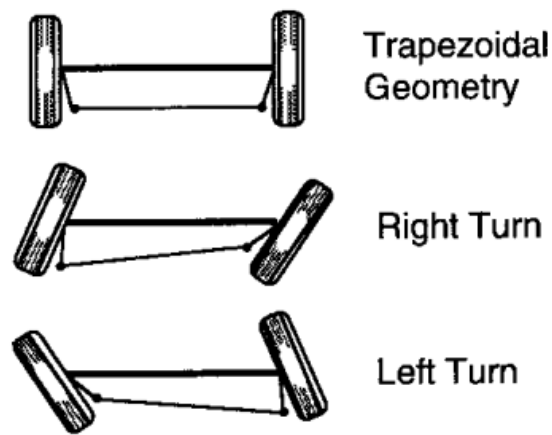


Figura 3: Trapecio de la dirección [1]

Para que el mecanismo de dirección sea efectivo debe cumplir la denominada regla de las cotangentes [1]. En términos de la Figura 2, esta condición tomaría la expresión:

$$\cot(\delta_o) - \cot(\delta_i) = \frac{t}{L} \quad [2.3]$$

2.1.2 Modelo del biciclo

Generalmente, para el desarrollo de algoritmos de control, el modelo de cuatro ruedas se simplifica en el “modelo de la bicicleta”. Esta simplificación consiste en sustituir las ruedas del eje trasero por una única rueda en la posición central del eje; igualmente las ruedas del eje delantero se sustituyen por una única rueda en la posición central, tal y como se representa en la Figura 4.

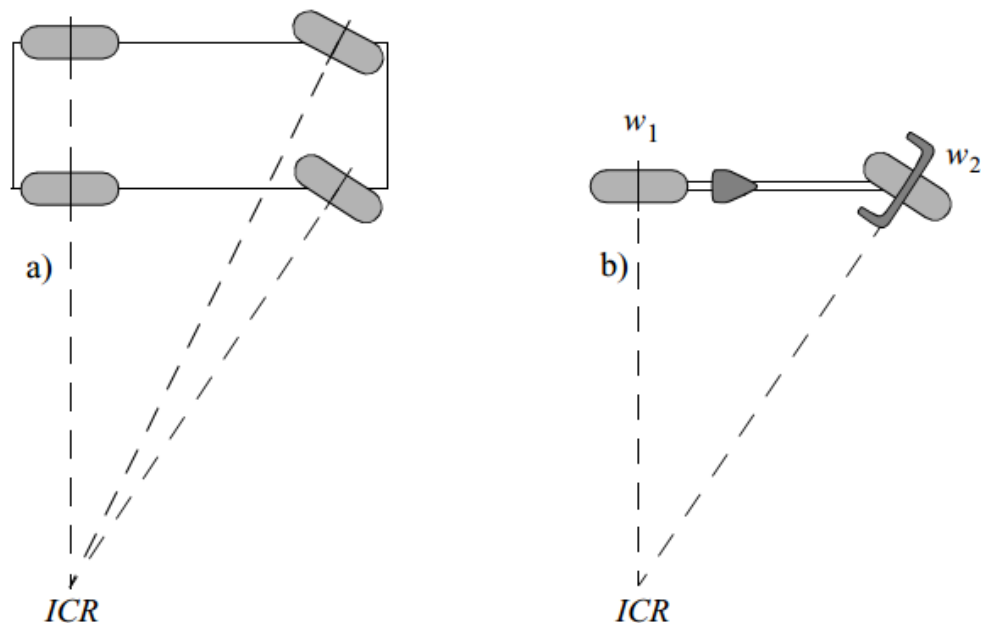


Figura 4: a) geometría Ackerman 4 ruedas; b) geometría Ackerman 2 ruedas [3]

La ventaja de aplicar esta simplificación es que, a la hora de hacer cálculos, solo hay que tener en cuenta el ángulo de una única rueda direccional; mientras que en el caso del cuadríciclo, como vimos antes, había que calcular el ángulo de la rueda exterior y de la interior.

Los parámetros geométricos del modelo quedan representados en la Figura 5. En esta figura se está haciendo además otra simplificación al decir que el ángulo δ coincide con el ángulo de dirección de la rueda delantera. Dichos ángulos coinciden cuando las fuerzas laterales aplicadas sobre el neumático son nulas, en caso contrario, la velocidad lineal de un punto cualquiera de la superficie del neumático no está alineada con el ángulo de dirección de la rueda, y por tanto los ángulos difieren. La suposición se considera válida para velocidades suficientemente bajas [1].

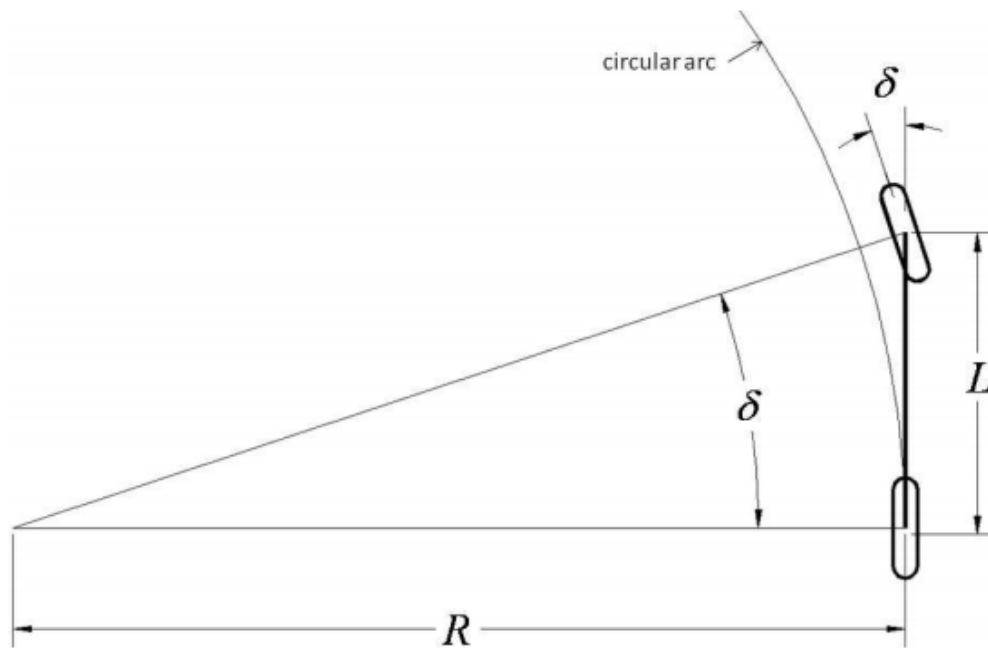


Figura 5: Modelo Ackerman, Bicicleta [2]

Dónde:

- "δ" es el ángulo de dirección que se aplica en la rueda directriz
- "L" es la distancia entre los ejes delantero y trasero del vehículo
- "R" es el radio de la curvatura que describe el vehículo.

Se deduce también la relación:

$$\tan(\delta) = \frac{L}{R} \quad [2.4]$$

El algoritmo de navegación que se desarrolla en este proyecto está basado en las relaciones geométricas entre el vehículo y la trayectoria a seguir. La simplificación del modelo de Ackerman al vehículo de dos ruedas es la que se utilizará en este proyecto para desarrollar el algoritmo de navegación.

En la solución propuesta se utilizará la ecuación 2.4 para calcular el ángulo de dirección a aplicar sobre el vehículo.

2.2 Métodos geométricos de “path tracking”

Dentro de los algoritmos de “path tracking” (seguimiento de trayectoria) destacan por su sencillez los modelos geométricos. El principio fundamental de estos modelos es establecer relaciones geométricas entre las características del vehículo y la vía o trayectoria a seguir, con la finalidad de obtener los comandos necesarios para la navegación del vehículo.

A continuación se desarrollan algunos de los modelos geométricos más extendidos: “follow the carot”, “pure pursuit” y “vector pursuit”. Una vez expuestos y analizados los tres métodos geométricos, se realiza una comparación entre ellos, a fin de justificar la elección para el algoritmo que se desarrolla en el capítulo 3.

2.2.1 Follow the carot

El modelo “follow the carot” es uno de los métodos geométricos de path tracking más antiguos y simples. La idea principal de este método consiste en calcular un punto objetivo (“carot point”) situado en la trayectoria deseada a una cierta distancia de la posición actual del vehículo (“look ahead distance”) [4].

Una vez obtenido el punto objetivo, se calcula el error de orientación, el cual se define como el ángulo que forma el vector “orientación actual” del vehículo (“heading”) con la recta que une el punto objetivo y el origen del sistema de referencia del vehículo. Todos estos conceptos quedan ilustrados en la Figura 6.

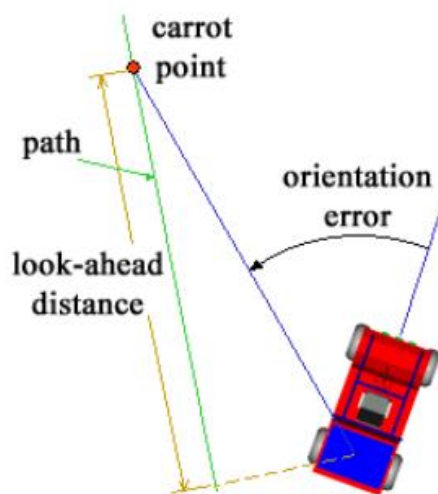


Figura 6: follow the carot [4]

La elección del parámetro “look ahead distance” tiene efectos determinantes en el resultado final del algoritmo [5]. Elegir una distancia muy pequeña provocará que el vehículo oscile en torno a la trayectoria deseada (“path” en la Figura 6).

Por el contrario, elegir una distancia grande hará que el vehículo tienda a eludir los giros bruscos (esquinas) que pudiera haber en la trayectoria deseada [4]; lo cual implica un mayor error de posición, definido como la diferencia entre la posición real del vehículo y la ideal; en ocasiones puede incluso suponer la colisión con un obstáculo. Los efectos comentados anteriormente se ilustran en la Figura 7.

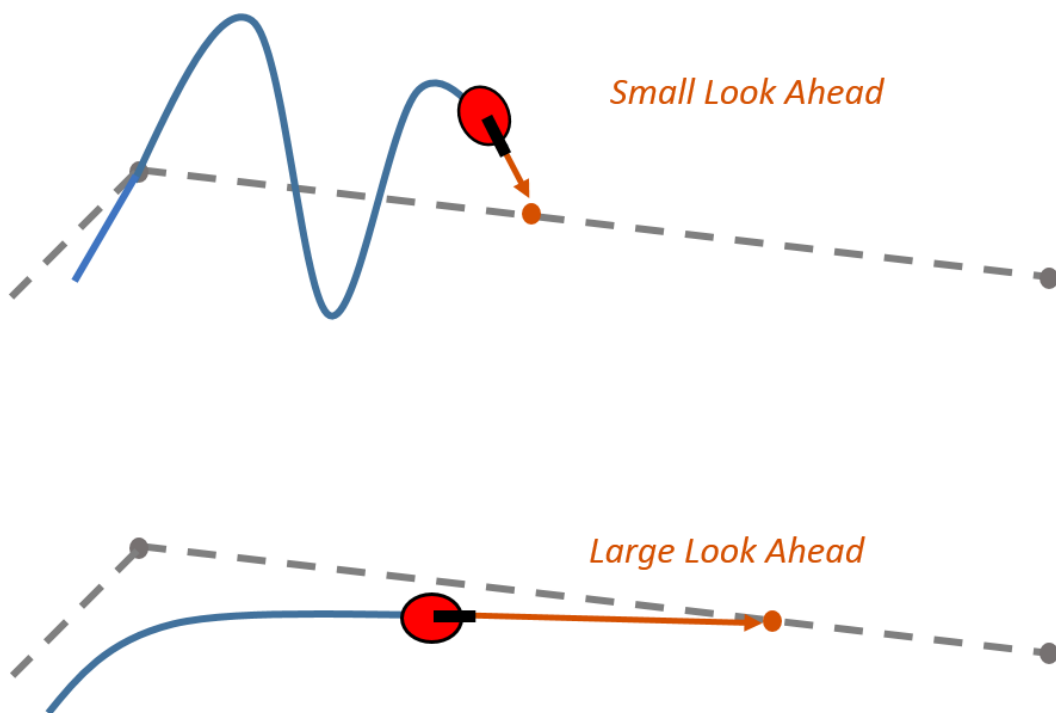


Figura 7: efectos en la elección del parámetro "look ahead distance" [5]

Además de la elección del parámetro “look ahead distance”, otro factor muy influyente es la velocidad del vehículo. Cuanto mayor es la velocidad, peor será la respuesta del algoritmo “follow the carrot”, ya que los efectos negativos de la elección del parámetro “look ahead distance” se acentúan [2].

Por todo lo comentado anteriormente, es de crucial importancia elegir un valor adecuado del parámetro “look ahead distance” según las características de la aplicación particular.

En la práctica el método “follow the carrot” tiene muchas limitaciones y por ello no se utiliza en ninguna aplicación que requiera un seguimiento preciso de la trayectoria deseada. No obstante por su sencillez y fácil implementación se utiliza con propósitos educativos o para comparar resultados con otros algoritmos [4].

2.2.2 Pure Pursuit

El método geométrico “Pure Pursuit” es probablemente el más extendido entre los algoritmos de “path tracking” basados en relaciones geométricas [2]. El modelo “Pure Pursuit” se basa en calcular los parámetros necesarios para que el vehículo describa una trayectoria circular (arco de circunferencia) [6]. Por tanto, el objetivo de este algoritmo es determinar el valor “ R ” de la Figura 8.

La filosofía del algoritmo “Pure Pursuit” consiste en calcular un punto objetivo dentro de una trayectoria dada, y posteriormente calcular los comandos necesarios para que el vehículo alcance dicho punto describiendo un arco de circunferencia. El punto objetivo es recalculado cada cierto intervalo de tiempo y siempre antes de que el vehículo lo alcance, metodología de la que proviene el nombre del algoritmo, ya que el vehículo se encuentra en una persecución constante del punto objetivo. Este comportamiento simula la conducción humana, ya que inconscientemente tendemos a fijar nuestra atención en un punto de la carretera y conducir hasta él [2].

El método “Pure Pursuit” comparte aspectos comunes con el modelo “follow the carot”. En ambos métodos se define el parámetro “look ahead distance” y en base a éste se calcula un punto objetivo en la trayectoria a seguir. Los elementos más característicos del modelo “Pure Pursuit” quedan ilustrados en la Figura 8.

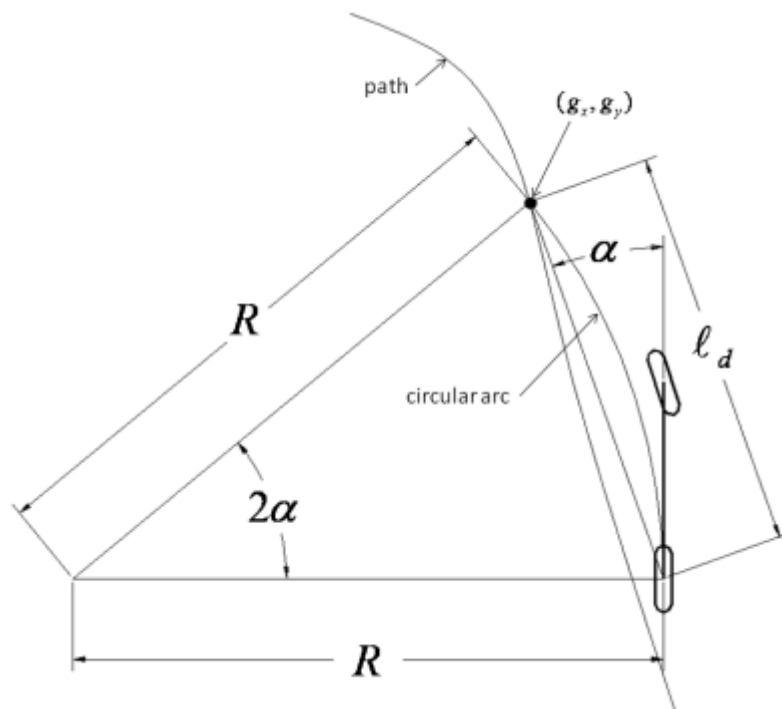


Figura 8: geometría “Pure Pursuit” [2]

Dónde:

- “ g_x ” y “ g_y ” son las coordenadas del punto objetivo (“goal point”) respecto al sistema de referencia del vehículo.
- “ α ” es el ángulo que forman los vectores “heading” y “goal” (ver Figura 9)
- “ l_d ” es el parámetro “look ahead distance”
- “ R ” es el radio de la curvatura que describe el vehículo.

Aplicando el teorema del seno sobre el triángulo de la Figura 8, se tiene:

$$\frac{l_d}{\sin(2\alpha)} = \frac{R}{\sin\left(\frac{\pi}{2} - \alpha\right)} \quad [2.5]$$

$$\frac{l_d}{2 \cdot \sin(\alpha) \cdot \cos(\alpha)} = \frac{R}{\cos(\alpha)} \quad [2.6]$$

Se deduce finalmente la relación:

$$R = \frac{l_d}{2 \cdot \sin(\alpha)} \quad [2.7]$$

Como ya se ha dicho, el modelo “Pure Pursuit” utiliza el parámetro “look ahead distance” para calcular un punto objetivo (“goal point”). Los efectos en la elección de dicho parámetro, que fueron analizados en la sección 2.2.1 (follow the carrot), son aplicables también al algoritmo “Pure Pursuit”.

En la Figura 9 se representa en azul el sistema de coordenadas del vehículo, así como los vectores “heading” y “goal” mencionados anteriormente. También sirve de apoyo al razonamiento sobre el valor del ángulo “ 2α ”.

Como se puede apreciar en la Figura 9, el triángulo formado es isósceles, dos de sus lados tienen magnitud “ R ” y el tercero “ l_d ”. Al trazar una perpendicular al vector “goal” el ángulo formado entre la perpendicular y la recta R es “ α ”. Si la perpendicular se traza por el vértice del triángulo isósceles (que es también el centro de la circunferencia) el triángulo queda dividido en dos triángulos rectángulos idénticos, por lo que se deduce que el valor del ángulo en el triángulo isósceles es “ 2α ”.

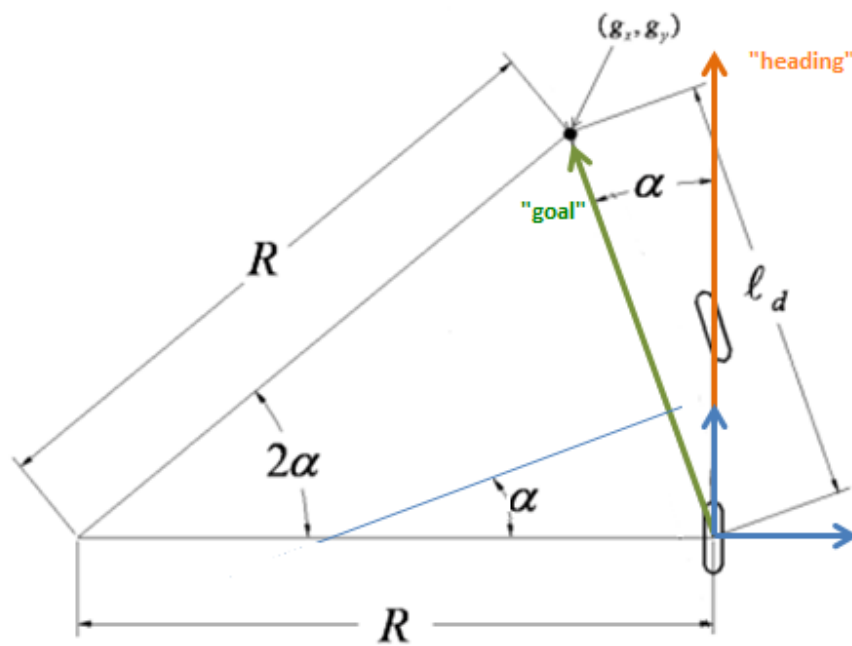


Figura 9: explicación parámetros "Pure Pursuit"

La implementación del método geométrico "Pure Pursuit" puede resumirse en los siguientes pasos [9]:

- 1) Determinar la posición actual del vehículo (posición y orientación).
- 2) Calcular un punto objetivo en la trayectoria deseada.
- 3) Calcular el radio o curvatura, que une la posición actual del vehículo con el punto objetivo. Teniendo en cuenta que éste se encuentra a una distancia "look ahead distance" de la posición actual del vehículo.
- 4) Calcular el ángulo de dirección necesario para que el vehículo describa la curvatura obtenida en el paso 3.
- 5) Actualizar la posición actual del vehículo.

Como aclaración, cabe resaltar que el paso 5 sólo es necesario en la fase de testeo del algoritmo, con la finalidad de realizar simulaciones de la trayectoria teórica que describirá el vehículo. En verdad el paso 1 y el paso 5 representan el mismo concepto, en la práctica, cuando se implementa el algoritmo Pure Pursuit, debe existir un sistema de localización que permita conocer en cada instante la posición y orientación del vehículo.

2.2.3 Vector Pursuit

El modelo “Vector Pursuit” es más completo que los anteriores, ya que permite tener un control no sólo de la posición, sino también de la orientación del vehículo. “Vector Pursuit” es un método geométrico basado en la “teoría de tornillos” en inglés “screw theory” [6]. Dicha teoría puede utilizarse para describir el movimiento de un sólido rígido respecto a un sistema de referencia dado. A continuación se explica brevemente en qué consiste la teoría de tornillos, se puede encontrar un análisis mucho más exhaustivo en la bibliografía [6] y [7], especialmente en [6] ya que ésta sección es un resumen del método presentado en dicho documento.

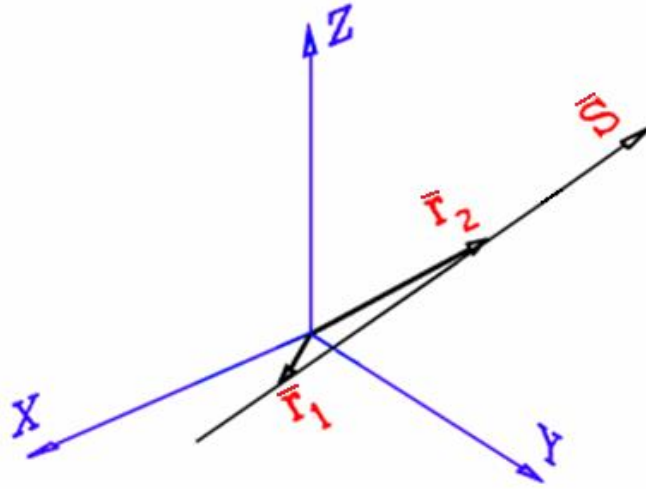


Figura 10: coordenadas Plücker, teoría de tornillos [6]

Un tornillo (“screw”) está definido por un eje de giro y un paso. Una forma de definir el eje de giro en el espacio tridimensional es utilizando las coordenadas de Plücker. Haciendo uso de la nomenclatura de la Figura 10, se tiene:

$$\bar{S} = \frac{\bar{r}_2 - \bar{r}_1}{|\bar{r}_2 - \bar{r}_1|} \quad [2.8]$$

$$\bar{S}_0 = \bar{r}_1 \times \bar{S} \quad [2.9]$$

El conjunto $(\bar{S}; \bar{S}_0)$ son las coordenadas de Plücker que definen el eje de giro del tornillo. Si se define $\bar{S} = (L, N, M)$, $\bar{S}_0 = (P, Q, R)$, $\bar{r}_1 = (x_1, y_1, z_1)$ y $\bar{r}_2 = (x_2, y_2, z_2)$ entonces, se pueden determinar los vectores \bar{S} y \bar{S}_0 según:

$$L = \frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \quad [2.10]$$

$$M = \frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \quad [2.11]$$

$$N = \frac{z_2 - z_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \quad [2.12]$$

$$P = y_1 N - z_1 M \quad [2.13]$$

$$Q = z_1 L - x_1 N \quad [2.14]$$

$$R = x_1 M - y_1 L \quad [2.15]$$

Utilizando ésta definición y estableciendo un valor para el “paso del tornillo” (en inglés “pitch”). La velocidad instantánea de cualquier punto de un sólido rígido, que rota con una velocidad angular “ ω ” en torno al tornillo “ $\bar{\$}$ ” cuyo eje de giro es “ $(\bar{S}; \bar{S}_0)$ ” y su paso es “ h ” se puede expresar según la ecuación:

$$\omega \bar{\$} = (\omega \bar{S}; \omega \bar{S}_{0h}) \quad [2.16]$$

Dónde:

$$\bar{S}_{0h} = \bar{S}_0 + h\bar{S} = \bar{r}_1 \times \bar{S} + h\bar{S} \quad [2.17]$$

Todos estos conceptos quedan reflejados en la Figura 11.

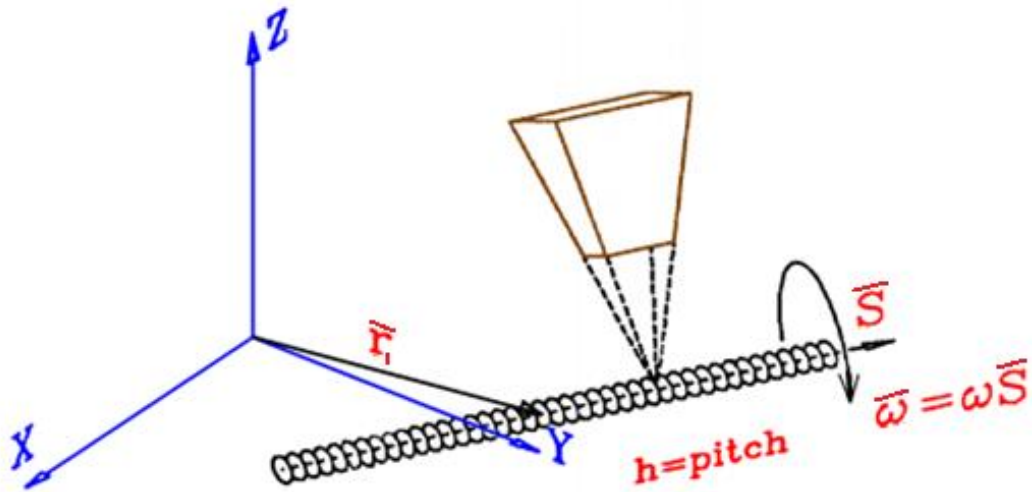


Figura 11: teoría de tornillos, movimiento del sólido rígido [6]

Cabe mencionar dos casos concretos que son de especial utilidad para el desarrollo del algoritmo “Vector Pursuit”. Si elegimos un tornillo cuyo paso es infinito, entonces el movimiento de un sólido rígido en torno a dicho tornillo, modela un movimiento de traslación puro, y la ecuación 2.10 que define dicho movimiento se simplifica a:

$$v \bar{\$} = (0; v \bar{S}) \quad [2.18]$$

Si por el contrario, elegimos un tornillo cuyo paso es nulo, entonces ese tornillo modela un movimiento rotacional puro, y la ecuación 2.10 se simplifica a:

$$\omega \bar{\$} = (\omega \bar{S}; \omega \bar{S}_0) \quad [2.19]$$

A continuación se presenta el desarrollo del algoritmo “Vector Pursuit” [6]. El método que se presenta no tiene en cuenta las restricciones no holónomas del vehículo, por lo que al final del cálculo se comprueba la validez de la solución obtenida.

A fin de realizar los cálculos necesarios, se definen tres sistemas de referencia tal y como se describen en la Figura 12.: un sistema de referencia fijo global (“world”); un sistema de referencia móvil adherido al vehículo (“vehicle”) cuyo origen es el punto central del eje trasero y cuyo eje X apunta siempre hacia la dirección de avance del vehículo; y por último un sistema de referencia móvil cuyo origen es el punto objetivo y cuyo eje X es coincidente con la trayectoria a seguir.

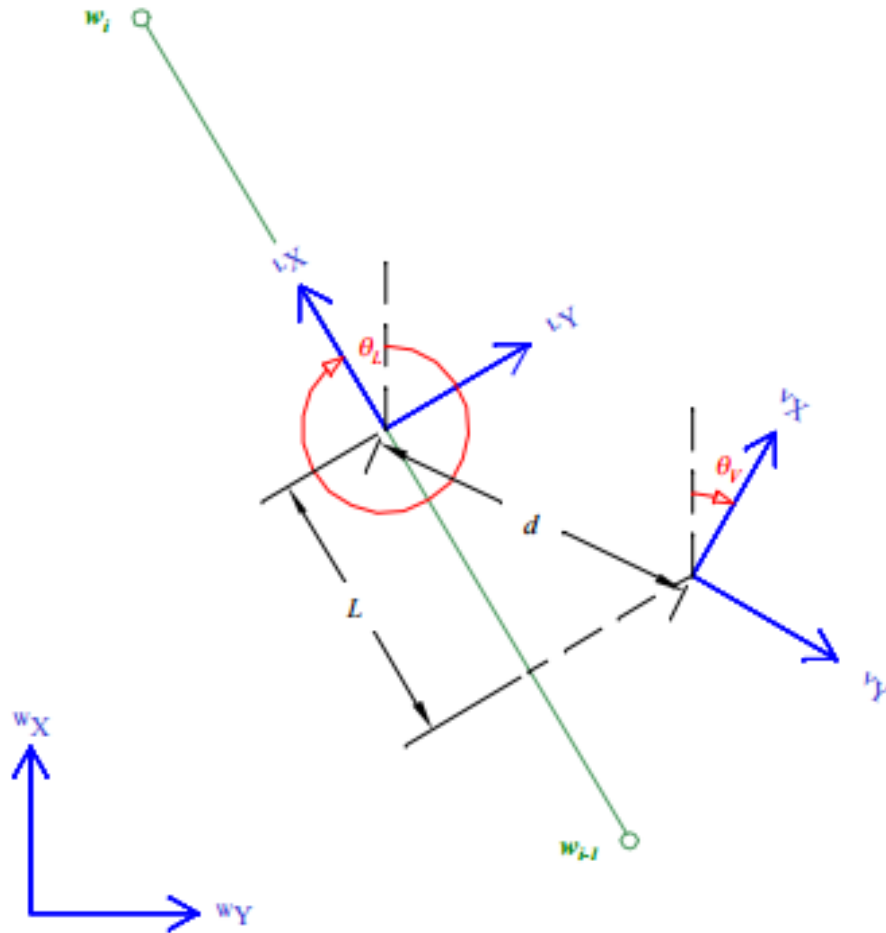


Figura 12: sistemas de referencia, vector pursuit [6]

El flujo del algoritmo empieza, como todos los métodos geométricos, con la elección de un valor para el parámetro “look ahead distance” a fin de obtener un punto objetivo, en la Figura 12 dicho valor es “L”. A continuación se calculan dos tornillos, uno será el encargado del movimiento traslacional $\bar{\$}_t$ y el otro del movimiento rotacional $\bar{\$}_r$. La suma de ambos tornillos $\bar{\$}_d$ da como resultado el tornillo que describe el movimiento deseado.

$$\bar{\$}_d = \bar{\$}_t + \bar{\$}_r \quad [2.20]$$

Utilizando la notación de la Figura 12, los tornillos de traslación y rotación toman las siguientes expresiones, respecto al sistema global (${}^wX, {}^wY$):

$$\bar{\$}_t = k_t \cdot \left(0, 0, 0; \frac{X_L - X_V}{d}, \frac{Y_L - Y_V}{d}, 0 \right) \quad [2.21]$$

$$\bar{\$}_r = k_r \cdot (0, 0, 1; Y_V, -X_V, 0) \quad [2.22]$$

Donde k_t y k_r son constantes no nulas que permiten aumentar o disminuir la influencia de cada movimiento, (X_L, Y_L) son las coordenadas del punto objetivo respecto al sistema global, y (X_V, Y_V) son las coordenadas del vehículo en el sistema global.

Sumando ambos tornillos, se obtiene el tornillo que describe el movimiento deseado:

$$\bar{\$}_d = \left(0, 0, k_r; k_r \cdot Y_V + k_t \cdot \frac{X_L - X_V}{d}, -k_r \cdot X_V + k_t \cdot \frac{Y_L - Y_V}{d}, 0 \right) \quad [2.23]$$

Para determinar el valor de las constantes k_t y k_r se definen t_t y t_r como el tiempo necesario para que el vehículo complete el movimiento de translación y el tiempo necesario para que complete el movimiento de rotación respectivamente.

$$t_t = \frac{d}{v} \quad [2.24]$$

$$t_r = \frac{\theta_L - \theta_V}{\omega} \quad [2.25]$$

Donde los ángulos θ_L y θ_V son los indicados en la Figura 12. Se establece además la relación:

$$t_r = k t_t \quad [2.26]$$

Observando la ecuación 2.23 cabe destacar que k_t es una velocidad lineal, y k_r es una velocidad angular. Teniendo esto en cuenta, y utilizando las ecuaciones 2.24, 2.25 y 2.26 se obtienen finalmente los valores de k_t y k_r .

$$K_t = v \quad [2.27]$$

$$k_r = \omega = \frac{\theta_L - \theta_V}{t_r} = \frac{\theta_L - \theta_V}{k \cdot t_t} = \frac{v(\theta_L - \theta_V)}{k \cdot d} \quad [2.28]$$

Una vez definidas ambas constantes, es necesario determinar, en el sistema de referencia del vehículo, la posición del eje de giro del tornillo $\bar{\$}_d$ de la ecuación 2.23. Para lo cual, se obtienen en primer lugar las coordenadas del eje de giro referidas al sistema global.

$${}^wX_{\bar{\$}_d} = X_V - \frac{k_t}{k_r} \left(\frac{Y_L - Y_V}{d} \right) = X_V - k \left(\frac{Y_L - Y_V}{\theta_L - \theta_V} \right) \quad [2.29]$$

$${}^wY_{\bar{\$}_d} = Y_V + \frac{k_t}{k_r} \left(\frac{X_L - X_V}{d} \right) = Y_V + k \left(\frac{X_L - X_V}{\theta_L - \theta_V} \right) \quad [2.30]$$

Una vez obtenidas las coordenadas del eje de giro respecto al sistema global, apoyándose en la geometría de la Figura 12, se obtiene las siguientes expresiones para las coordenadas del eje de giro respecto al sistema de referencia del vehículo.

$${}^VX_{\bar{\$}_d} = X_V \cdot \cos(\theta_V) + Y_V \cdot \sin(\theta_V) - ({}^wX_{\bar{\$}_d} \cdot \cos(\theta_V) + {}^wY_{\bar{\$}_d} \cdot \sin(\theta_V)) \quad [2.31]$$

$${}^VY_{\bar{\$}_d} = Y_V \cdot \cos(\theta_V) - X_V \cdot \sin(\theta_V) - ({}^wY_{\bar{\$}_d} \cdot \cos(\theta_V) - {}^wX_{\bar{\$}_d} \cdot \sin(\theta_V)) \quad [2.32]$$

Estas coordenadas corresponden con la posición del centro de una de las circunferencias que se representan en la Figura 13. No obstante, debido a las restricciones no holónomas, teniendo en cuenta la orientación actual del vehículo, sólo la circunferencia marcada en rojo en la Figura 13 es una posible trayectoria. El método descrito hasta ahora puede dar como resultado las coordenadas del centro correcto o del centro incorrecto, por tanto es necesario calcular el otro centro y determinar cuál de ellos es el correcto. Como se aprecia en la Figura 13, el centro correcto será aquel cuya coordenada X en el sistema de referencia del vehículo sea nula.

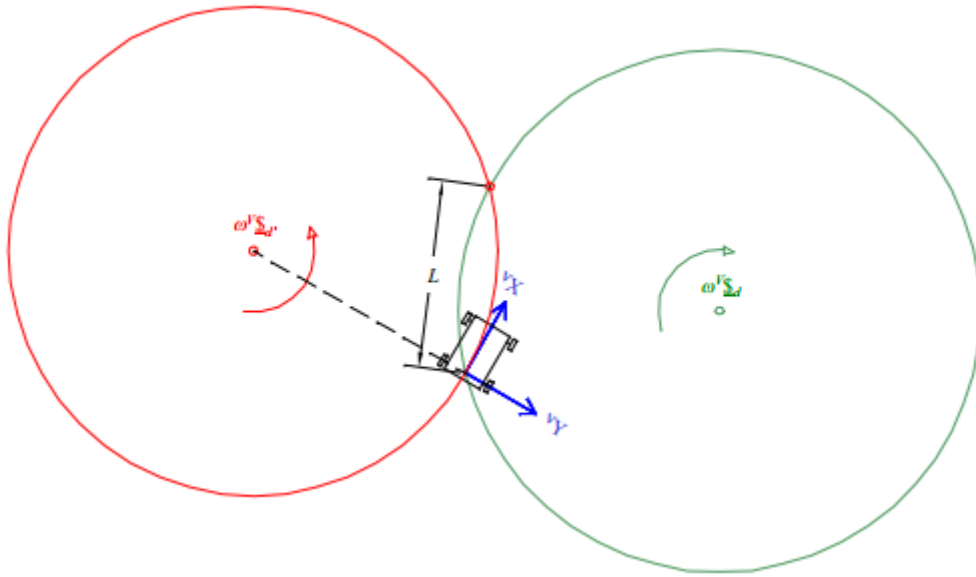


Figura 13: circunferencias, vector pursuit [6]

Una vez obtenido el centro correcto, se calcula el radio de giro, el cual se utiliza finalmente para calcular el ángulo de dirección necesario. Utilizando la Figura 14 se Pueden plantear las siguientes ecuaciones:

$$a^2 + v x_p^2 = R^2 \quad [2.33]$$

$$a = R - v y_p \quad [2.34]$$

$$v x_p^2 = L^2 - v y_p^2 \quad [2.35]$$

Sustituyendo y resolviendo para R:

$$R = \frac{L^2}{2 v y_p^2} \quad [2.36]$$

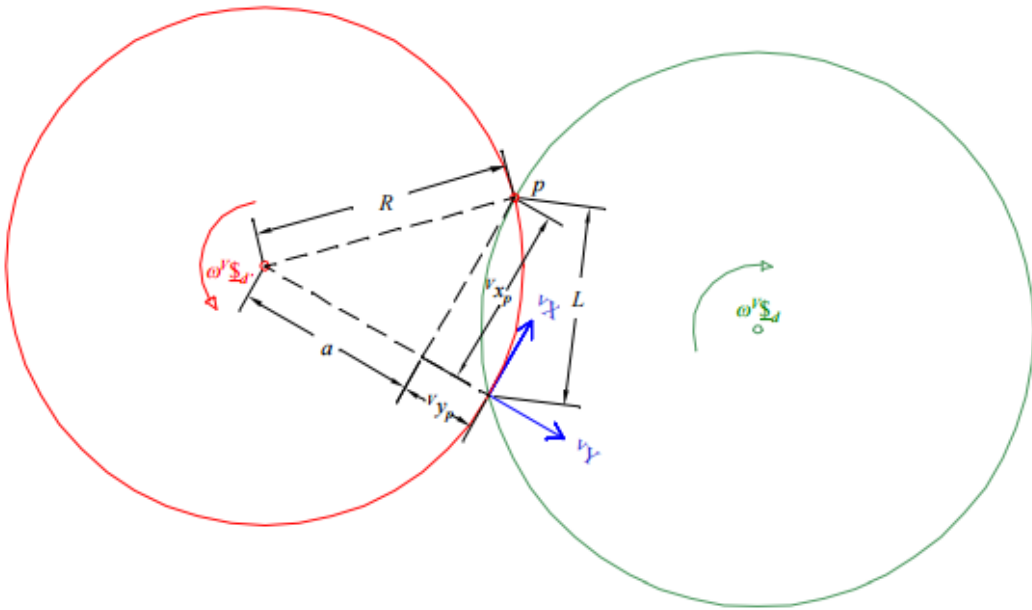


Figura 14: cálculo del radio, Vector Pursuit [6]

Ecuación 2.36 que es muy similar a la ecuación 2.7 de la geometría Pure Pursuit.

2.2.4 Comparación entre los métodos geométricos

Una vez presentado el desarrollo teórico de los distintos métodos geométricos de “path tracking” que se han analizado, se pueden comparar entre sí para decidir cuál de ellos se ajusta más a la aplicación de este proyecto. En la Tabla 1 se presenta a modo de resumen las características principales de cada uno de los algoritmos.

	posicionamiento	orientación	complejidad	resultados
Follow the carrot	Si	No	Baja	Insuficiente según la aplicación
Pure Pursuit	Si	No	Media	aceptable
Vector Pursuit	Si	Si	Alta	aceptable

Tabla 1: Resumen algoritmos Path Tracking

En cuanto a la implementación del algoritmo, queda claro que el método “Vector Pursuit” es el más complejo de los tres algoritmos expuestos, tanto por la teoría en que está basado, como en su implementación. No obstante es también el único de los tres métodos analizados, que tiene en cuenta la orientación del vehículo en el punto objetivo, los otros dos métodos, “follow the carrot” y “Pure Pursuit” sólo consideran el posicionamiento.

Utilizar un algoritmo que considere la orientación del vehículo en el punto objetivo, puede resultar especialmente útil para aquellos vehículos cuyo movimiento no este restringido únicamente a la dirección de avance, como por ejemplo los vehículos de locomoción diferencial o los vehículos de tres ruedas síncronas [10]. En este tipo de configuraciones, sí tiene sentido considerar la orientación en el punto objetivo, ya que al no presentar restricciones holónomas, es posible establecer distintas trayectorias entre el punto actual y el punto objetivo.

No obstante en los vehículos con configuración Ackerman (tipo cuadríciclo), el control de la orientación no es tan crítico, ya que una vez elegido el punto objetivo sólo es posible llegar hasta él mediante una única trayectoria, es decir, la orientación final está determinada por la elección del punto objetivo.

Por supuesto, diseñar un algoritmo de navegación para un vehículo tipo Ackerman, que tenga en cuenta la orientación del vehículo en el punto objetivo, es posible. No obstante, o bien el algoritmo de planificación de trayectorias (el encargado de definir los waypoints), o bien el propio algoritmo de navegación, tendrían que decidir en cada movimiento, con qué orientación debe quedar el vehículo. Lo cual implica una filosofía de trabajo diferente a las planteadas, ya que el parámetro “look ahead distance” carecería de sentido, puesto que la condición para obtener un punto objetivo dependería de la orientación con la que se desea terminar el movimiento.

Por todo lo comentado, la implementación más razonable en este caso particular, es considerar únicamente el posicionamiento, puesto que los vehículos para los que se pretende diseñar el algoritmo de navegación en este proyecto, son de configuración Ackerman. Entre los algoritmos “follow the carrot” y “Pure Pursuit”, como se comentó en las secciones 2.2.1 y 2.2.2, los resultados ofrecidos por éste último son superiores, razón por la cual “Pure Pursuit” es el algoritmo de navegación que se desarrolla en el presente proyecto.

2.3 Planteamiento del problema

Al momento de empezar a trabajar en este trabajo fin de grado, ya existían dentro del proyecto iCab algoritmos de planificación de trayectorias. Es decir, el entorno de navegación (el campus de la universidad) ya estaba modelado como un mapa de cuadrículas y ya existía un algoritmo capaz de encontrar la ruta óptima partiendo de un origen y eligiendo un destino.

En la Figura 15 se puede ver una ruta de ejemplo obtenida mediante dicho código. El punto inicial está marcado en verde, y el destino está marcado en rojo. La misma ruta se representa en la Figura 16, pero en este caso se pueden apreciar los puntos (waypoints) que la forman, los cuales se almacenan en un fichero de datos.

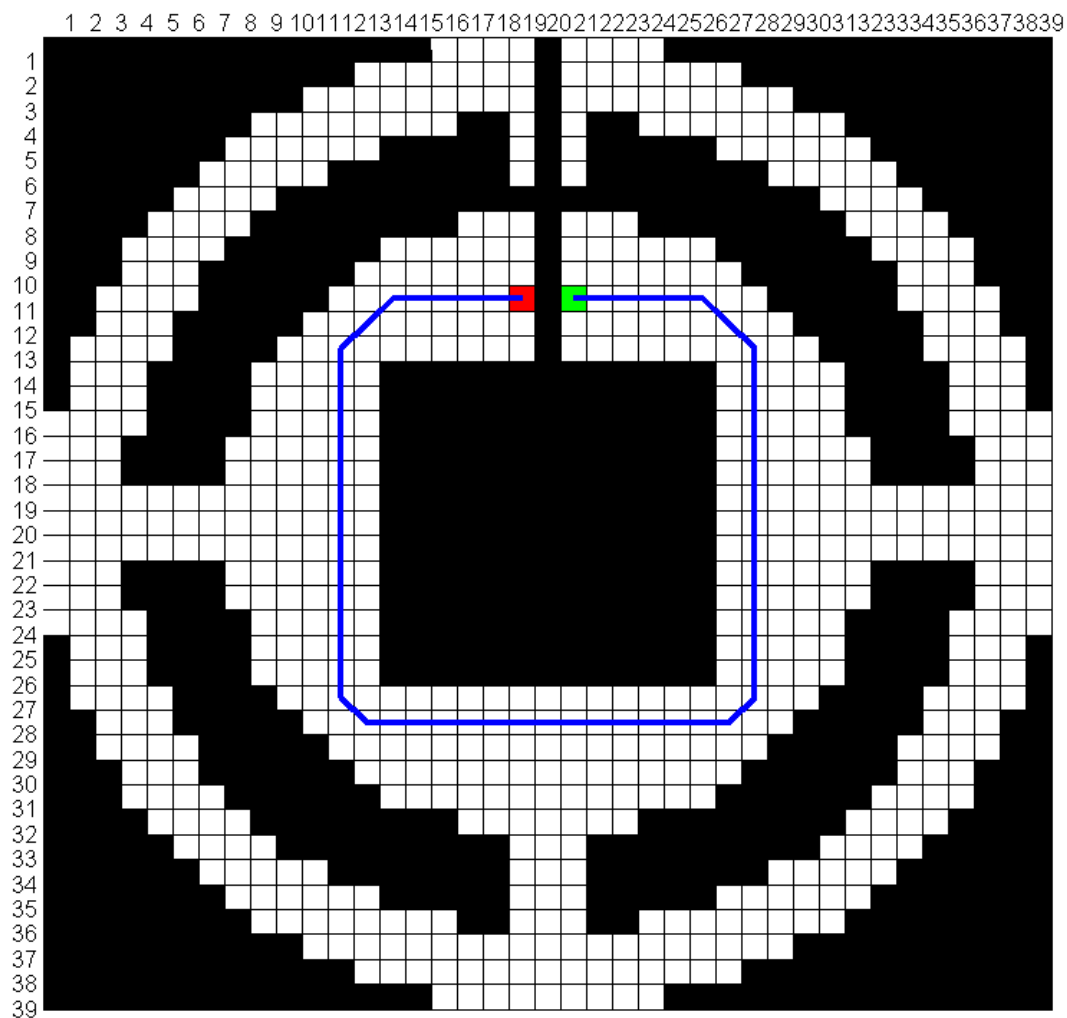


Figura 15: Ejemplo ruta de calculada

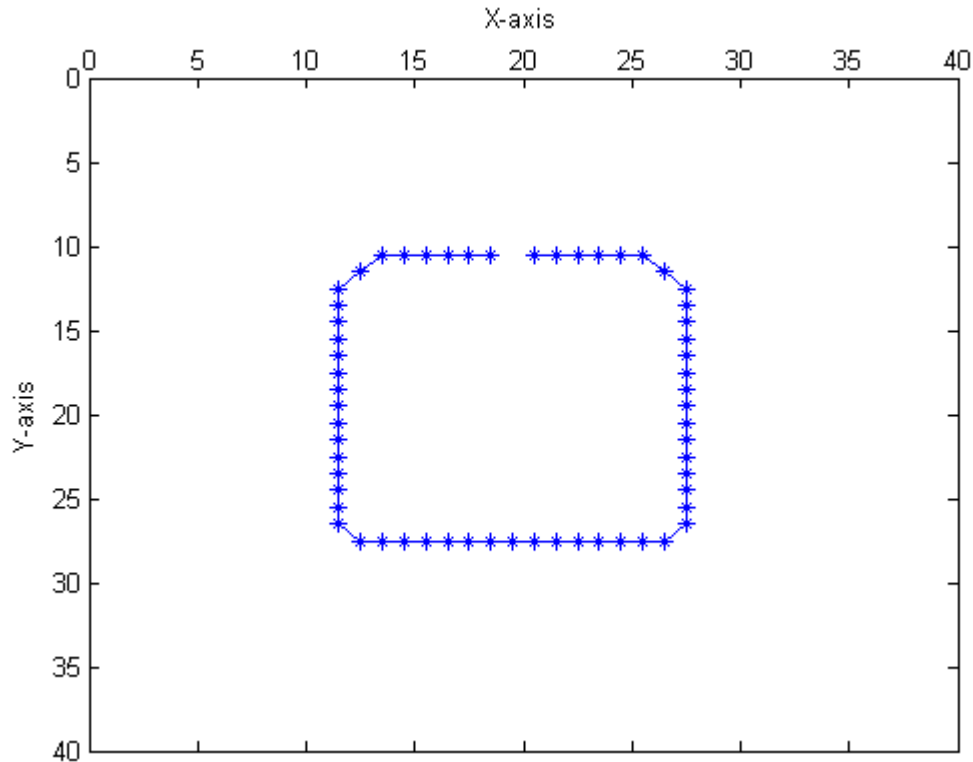


Figura 16: ejemplo de waypoints

No obstante, la ruta ofrecida por dicho algoritmo no es un camino navegable por el vehículo. Es decir, los waypoints no se pueden considerar puntos objetivo y calcular los comandos a introducir en el vehículo para que siga la ruta. Esto es debido a dos razones:

En primer lugar, a fin de conseguir una mayor precisión los puntos están separados entre sí una distancia de un metro, si la línea es horizontal o vertical y una distancia de $\sqrt{2}$ metros si la línea es oblicua. El hecho de que los puntos estén tan próximos entre sí, en comparación con la longitud del vehículo (1,6 metros) hace que sea imposible considerar los waypoints como puntos objetivo.

En segundo lugar, el vehículo presenta restricciones holónomas, es decir solo puede desplazarse “hacia adelante”, no puede rotar sobre sí mismo sin alterar su posición. Esto es común a todos los vehículos cuya dirección es de tipo Ackerman. Por tanto, cabe la posibilidad de que al llegar a un determinado punto con una cierta orientación, no sea posible alcanzar el siguiente, especialmente cuando se quiera hacer un giro, ya que como se aprecia en la Figura 16, el ángulo entre dos puntos es de 0° ó de 45° .

Es necesario por tanto, programar un algoritmo que resuelva el problema de los waypoints. La solución propuesta consta de dos partes. Primero se trabaja sobre los waypoints a fin de crear una trayectoria a partir de los puntos, y por último se aplica el algoritmo “pure pursuit” junto con el modelo del biciclo de Ackerman, para obtener los comandos necesarios para la navegación, es decir, el ángulo de dirección a aplicar en cada instante (“steering angle”).

///

Solución propuesta

3.1 Obtención de la trayectoria a partir de los waypoints

El primer paso antes de poder aplicar el algoritmo Pure Pursuit, es obtener una trayectoria a la que aproximarse, a fin de poder calcular un punto objetivo dentro de dicha trayectoria. Los datos de entrada que recibe el algoritmo son una lista de puntos consecutivos referenciados al sistema de referencia global, en coordenadas cartesianas (x,y). Dichos puntos reciben el nombre de waypoints, ya que indican el camino que debe seguir el vehículo.

Por consiguiente, a partir de los waypoints hay que construir una trayectoria. Matemáticamente, hay que encontrar una función que se ajuste lo máximo posible a los waypoints. Este problema se conoce como “ajuste de curvas” o en inglés “curve fitting”. Existe una gran cantidad de métodos para ajustar una curva o función matemática a partir de una serie de puntos, desde los más sencillos como la regresión lineal, hasta otros más complejos como las “splines” o el ajuste por polinomios de alto grado. El método más efectivo dependerá de la aplicación concreta, es decir, de cómo sea la distribución de los puntos, pero en general será el más sencillo de entre todos aquellos que consigan un ajuste óptimo.

Fijando la atención en la Figura 15, se puede apreciar que el código mediante el cual se obtienen los waypoints tiende a generar trayectorias rectas. Parece por tanto razonable utilizar algún tipo de regresión lineal sobre una cantidad finita de puntos en la dirección de avance. Eso es precisamente lo que se ha hecho en el presente proyecto, el método de regresión lineal elegido ha sido el conocido método de mínimos cuadrados, utilizando como datos un determinado número de waypoints a partir de la posición actual del vehículo. A continuación se presenta una breve revisión del método de mínimos cuadrados.

3.1.1 Mínimos cuadrados

Una de las aplicaciones del método de mínimos cuadrados es obtener una función lineal a partir de una serie de datos, los cuales son parejas de valores (x,y). Comúnmente este método es utilizado para establecer relaciones lineales entre dos variables a partir de datos empíricos.

El resultado de la aplicación del método será por tanto una función de la forma:

$$y = m \cdot x + n \quad [3.1]$$

Para determinar los parámetros m y n que definen la función, se hace uso de las siguientes expresiones:

$$m = \frac{N \cdot \sum xy - \sum x \cdot \sum y}{N \cdot \sum x^2 - \sum x \sum x} \quad [3.2]$$

$$n = \frac{\sum y - m \cdot \sum x}{N} \quad [3.3]$$

Dónde:

· N es el número de datos (número de waypoints que definen la recta)

$$\begin{aligned}\sum x y &= \sum_{i=1}^N x_i \cdot y_i & \sum x &= \sum_{i=1}^N x_i \\ \sum x x &= \sum_{i=1}^N x_i \cdot x_i & \sum y &= \sum_{i=1}^N y_i\end{aligned}$$

Observando la ecuación 3.2 cabe destacar que existe la posibilidad de que al operar se dé el caso “división entre cero”. Es necesario por tanto analizar esta ecuación para realizar una operación alternativa cuando el denominador de la expresión sea nulo.

El denominador de la ecuación 3.2 se anula cuando:

$$N \cdot \sum_{i=1}^N x_i^2 = \left(\sum_{i=1}^N x_i \right)^2 \quad [3.4]$$

$$N \cdot (x_1^2 + x_2^2 \dots + x_N^2) = (x_1 + x_2 + \dots + x_N)^2 \quad [3.5]$$

Esta igualdad sólo se cumple si todos los términos x_i son iguales, es decir:

$$N \cdot \sum_{i=1}^N x_i^2 = \left(\sum_{i=1}^N x_i \right)^2 \quad \text{si y sólo si} \quad x_1 = x_2 = \dots = x_N$$

En cuyo caso, la ecuación de la función lineal correspondería con una recta vertical, de la forma:

$$x = x_i$$

Si por el contrario el denominador de la ecuación 3.2 no se anula, la función resultante corresponderá a una recta horizontal u oblicua y los parámetros m y n se calculan según las expresiones 3.2 y 3.3

3.1.2 Ecuación de la recta

Una vez obtenida la ecuación de la recta en su forma explícita (ecuación 3.1), se despeja la expresión para obtener la ecuación implícita de la recta. El motivo de trabajar con la ecuación implícita es que de cara a la programación es más sencillo trabajar con tres parámetros A, B, C ya que de esta manera, es fácil identificar el tipo de recta, lo cual resulta útil para resolver los sistemas de ecuaciones que se analizarán más adelante. La ecuación con la que se trabaja es de la forma:

$$Ax + By + C = 0 \quad [3.6]$$

De esta manera:

Si $A \neq 0$ y $B \neq 0$ entonces, la recta es oblicua

Si $A = 0$ entonces, la recta es horizontal $y = \frac{-C}{B}$

Si $B = 0$ entonces, la recta es vertical $x = \frac{-C}{A}$

Partiendo de la ecuación 3.1 se pueden establecer las siguientes relaciones entre los parámetros m y n que se calculaban por el método de mínimos cuadrados, y los parámetros A, B, C de la ecuación 3.6:

Recta oblicua

$$y = m \cdot x + n$$

$$m \cdot x - y + n = 0$$

Comparando ésta última expresión con la ecuación 3.6 se deduce:

$$A = m \quad ; \quad B = -1 \quad ; \quad C = n$$

Recta horizontal ($m = 0$)

$$y = n$$

De la ecuación 3.3 se obtiene valor de n para $m = 0$

$$n = \frac{\sum y}{N}$$

Luego, la ecuación de la recta será:

$$\frac{\sum y}{N} - y = 0$$

Comparando ésta última expresión con la ecuación 3.6 se deduce:

$$A = 0 \quad ; \quad B = -1 \quad ; \quad C = n = \frac{\sum y}{N}$$

Recta vertical ($m \rightarrow \infty$)

$$x = k \quad ; \quad k \in \mathbb{R}$$

Como ya se razonó en la sección 3.2.1 el valor de la constante k es x_i la ecuación de la recta será por tanto:

$$x_i - x = 0$$

Comparando ésta última expresión con la ecuación 3.6 se deduce:

$$A = -1 \quad ; \quad B = 0 \quad ; \quad C = x_i$$

Los resultados de todos los desarrollos anteriores, quedan resumidos en la Tabla 2

Recta	A	B	C
Oblicua	m	-1	n
Horizontal	0	-1	n
vertical	-1	0	x_i

Tabla 2: resumen del valor de los parámetros de la recta

Hasta ahora se ha explicado cómo obtener la ecuación de una recta a partir de un determinado número de waypoints. No obstante queda por explicar el funcionamiento del algoritmo que se encarga de obtener la trayectoria a la que aproximarse en cada movimiento. A continuación se detalla la implementación de dicho algoritmo.

Antes de efectuar cada movimiento, se calcula una nueva trayectoria a la que aproximarse, que como ya se ha dicho corresponde con la ecuación de una recta. Para definir los parámetros A,B,C de la ecuación 3.6 se aplica el método de mínimos cuadrados sobre 8 waypoints, que sean los más cercanos al vehículo en la dirección de avance.

En la Figura 17, se representa un camino de ejemplo para visualizar con mayor claridad la idea presentada. El punto marcado en naranja es la posición actual del vehículo, en azul se representan los waypoints y con la línea discontinua se indica la recta calculada, a la cual se aproxima el vehículo. Los puntos marcados en verde son los que se están utilizando para definir la recta en cada caso.

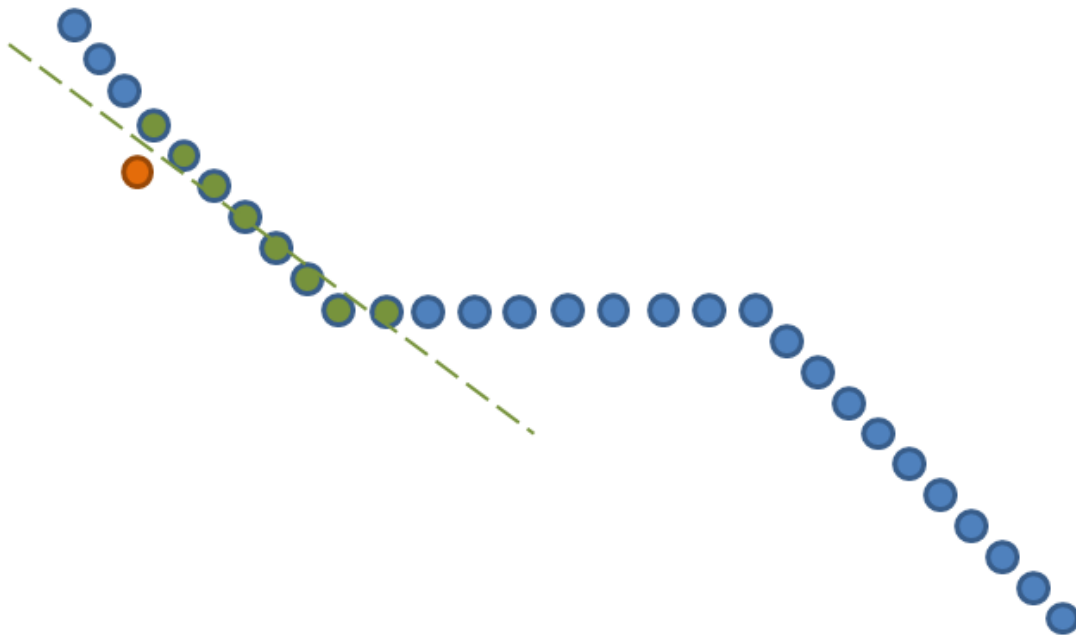


Figura 17: ejemplo recta de aproximación

Incluir el último waypoint que el vehículo haya sobrepasado, permite incluir información de la trayectoria anterior a la hora de calcular la nueva trayectoria. Esto logra generar una recta que se ajuste más a los waypoints y que a la vez genere una trayectoria del vehículo más suavizada.

La razón de elegir ocho y no otro número es que la distancia entre waypoints es originalmente de un metro. Por tanto, al utilizar ocho waypoints se está escogiendo un tramo de entre ocho y diez metros entre la posición actual del vehículo y el último waypoint. Los resultados de las simulaciones realizadas mostraron que dicha elección era adecuada.

Los criterios que se apliquen para seleccionar los siguientes waypoints que definirán la próxima recta, influirán de manera decisiva en el resultado final del algoritmo, ya que el vehículo se comanda para aproximarse a la trayectoria marcada por dicha recta.

El algoritmo desarrollado selecciona los waypoints en base a su proximidad a la posición actual del vehículo. Lo cual, a priori, podría suponer un problema cuando los waypoints formen un círculo o cualquier otra figura cerrada, o bien si hay intersecciones entre los waypoints, por ejemplo en un camino de ida y vuelta.

Por tanto es estrictamente necesario que los waypoints estén dados en orden, es decir, que tengan un índice. De cara a la programación esto no es realmente un problema, ya que los waypoints están almacenados en un fichero, y al hacer la lectura de dicho fichero se les está asignado un índice.

A modo de ejemplo, en la Figura 18 se representa el resultado real del algoritmo tomando como camino de ejemplo el establecido en la Figura 16. Dado que la finalidad es simplemente ilustrar el resultado, por claridad en el dibujo, no se representan todas las rectas necesarias completar el camino. Las rectas que se representan están numeradas del 1 al 6 indicando su orden.

En la Figura 18 se representan en color azul los waypoints. Los puntos representados en color verde son los “puntos objetivo” que calcula el algoritmo, cuya obtención se explica en la siguiente sección. La razón por la que se ha decidido incluirlos en la Figura 18 es que el lector aprecie que los puntos objetivo calculados pertenecen a cada una de las rectas obtenidas.

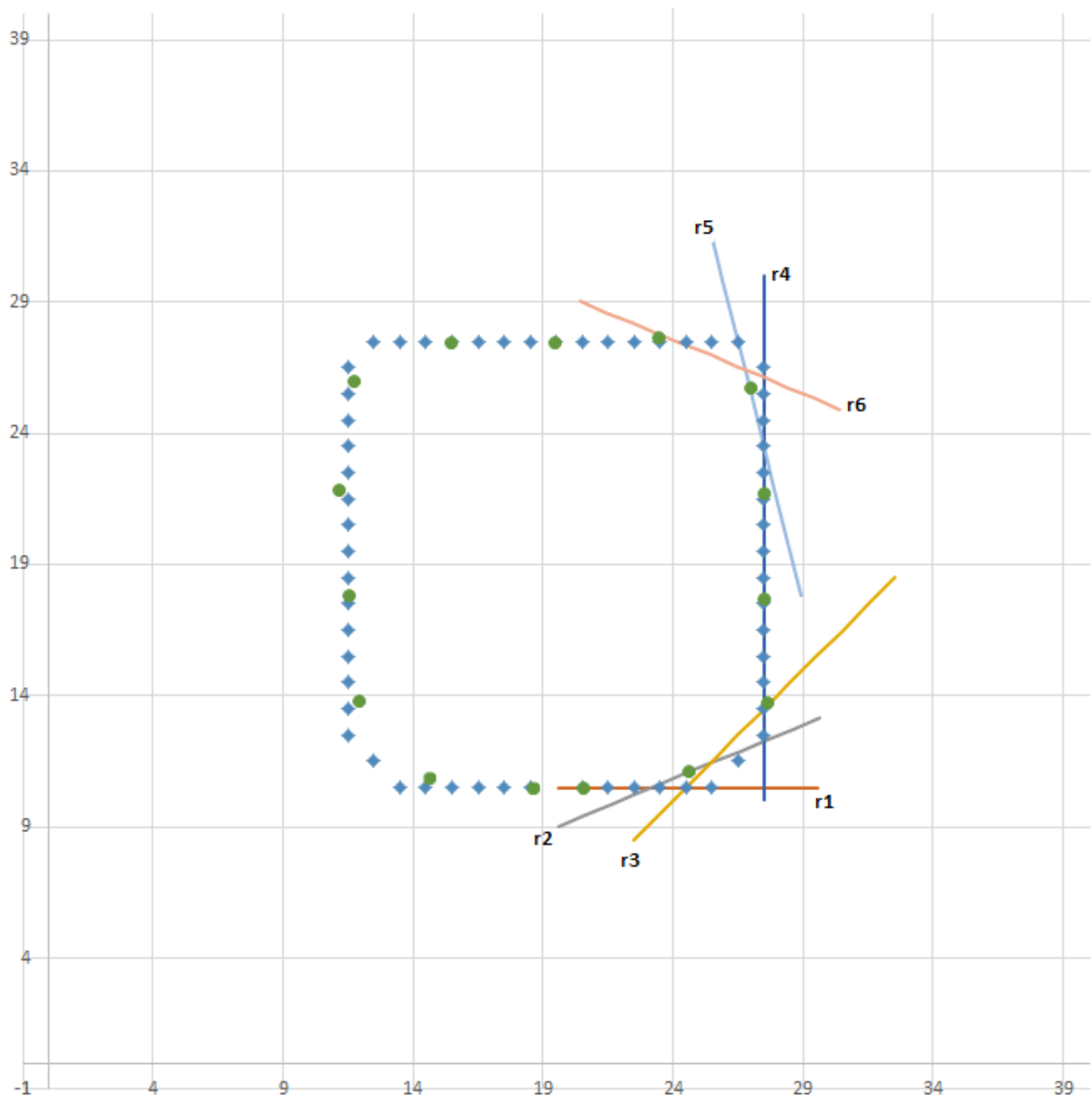


Figura 18: ejemplo generación de trayectoria, rectas consecutivas

3.2 Implementación del algoritmo Pure Pursuit

A continuación se proceden a explicar los detalles de la implementación del algoritmo Pure Pursuit, cuyo fundamento teórico se presentó en la sección 2.2.2 de este documento.

El primer paso para desarrollar el algoritmo es calcular un punto objetivo dentro de la trayectoria previamente obtenida. Para ello es necesario definir el parámetro look ahead distance. Las implicaciones relacionadas con la elección de dicho parámetro se presentaron en la sección 2.2.1, no obstante se presenta a continuación un breve resumen.

3.2.1 Elección del parámetro “look ahead distance”

Aumentar el valor del parámetro “look ahead distance” implica calcular un punto objetivo más lejano a la posición actual del vehículo, lo cual se traduce en trayectorias más suavizadas pero menos fieles al camino deseado. Por el contrario, disminuir el valor del parámetro implica calcular un punto objetivo más próximo a la posición actual del vehículo, la trayectoria será más fiel al camino original pero más abrupta. Este hecho se ilustra con claridad en la Figura 7 de la sección 2.2.1.

Por lo tanto, elegir un valor adecuado para el parámetro “look ahead distance” es de vital importancia para conseguir un buen resultado final. A fin de elegir un valor adecuado para éste parámetro, hay que tener en cuenta las características del vehículo, principalmente:

- La velocidad media a la que se moverá el vehículo.
- El radio de giro mínimo

En cuanto a la velocidad media, en la hoja de especificaciones del vehículo (ver Anexo 1) se puede ver que la velocidad máxima del mismo es de 21 *km/h*. Por lo que se considera que la velocidad media de la aplicación será de unos 18 a 20 *km/h*. Como se analizó en la sección 2.2.1, el efecto de la velocidad influye en la elección del parámetro “look ahead distance”. A mayor velocidad es necesario aumentar el valor del parámetro para conseguir buenos resultados.

No obstante, en la aplicación que ocupa el presente proyecto, la velocidad de trabajo se puede considerar baja ya que en el peor de los casos, el vehículo alcanzaría los 21 *km/h*. Por lo tanto es posible ajustar un valor “a la baja” para el parámetro “look ahead distance”.

Respecto al radio mínimo de giro, es también una de las restricciones a tener en cuenta una vez calculado el ángulo de dirección que se comanda al vehículo. Es decir, si el ángulo necesario para dirigir el vehículo hacia el punto objetivo, es mayor que el máximo ángulo que permite el sistema de dirección, la solución calculada no es válida, y es preciso recalcular el punto objetivo.

El radio mínimo de giro, es una definición teórica que suele darse como parámetro de los vehículos con configuración Ackerman. Dicho parámetro hace referencia al mínimo radio de la circunferencia que describe el vehículo cuando se gira por completo la dirección y se mantiene a lo largo del tiempo.

En la Figura 19 se puede ver que el valor de éste parámetro en los vehículos utilizados es de 2.9 metros, lo cual se traduce en un ángulo máximo de dirección de 30° en sentido horario y de -30° en sentido antihorario. Valores entre los que deberá encontrarse el ángulo de dirección calculado por el algoritmo para considerarlo válido.

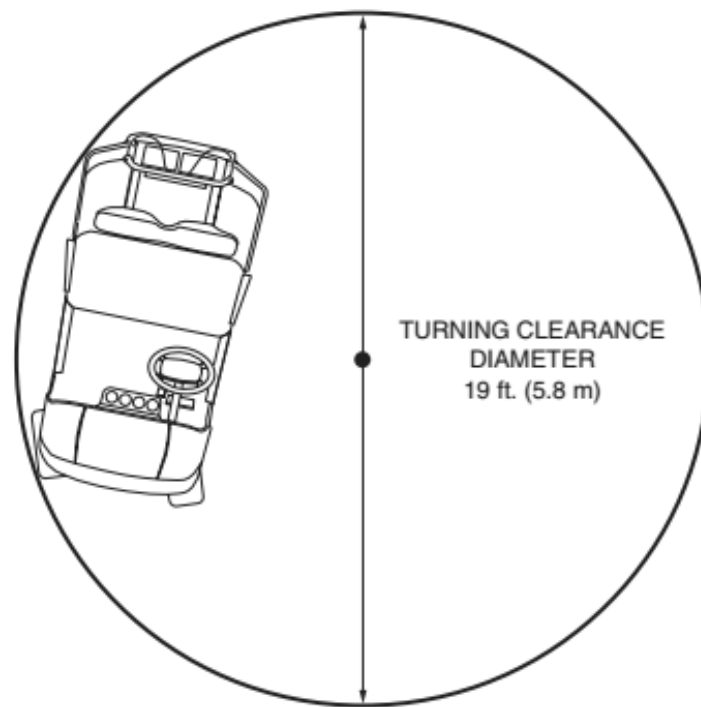


Figura 19: radio mínimo de giro [8]

Teniendo en cuenta todo lo comentado hasta ahora, se hicieron simulaciones con distintos valores para el parámetro “look ahead distance”, obteniéndose los mejores resultados teóricos con un valor de 4 metros.

Los resultados de dichas simulaciones se presentan con mayor detalle en la sección 4.1 de este documento.

3.2.2 Cálculo del punto objetivo

Una vez calculada la trayectoria a la que el vehículo debe aproximarse y definido el parámetro “look ahead distance”, es posible obtener el punto objetivo hacia el que comandar el vehículo.

Para ello, se obtiene en primer lugar la proyección ortogonal del origen del sistema de coordenadas del vehículo sobre la recta calculada. Matemáticamente se calcula la ecuación de la recta que pasa por el origen del sistema de coordenadas del vehículo (V_x, V_y) y cuyo vector director es perpendicular a la recta de aproximación. Lo cual se traduce en resolver el siguiente sistema de ecuaciones, cuya solución (x, y) será la proyección del vehículo sobre la recta de aproximación:

$$\begin{cases} Ax + By + C = 0 \\ y = \frac{B}{A}(x - V_x) + V_y \end{cases} \quad [3.7]$$

Resolviendo el sistema 3.7 para la componente x se obtiene:

$$x = \frac{B^2 V_x - A(B V_y + C)}{A^2 + B^2} \quad [3.8]$$

Una vez obtenido x se resuelve para la componente y :

$$y = \frac{-C - Ax}{B} \quad [3.9]$$

Ya calculada la proyección del origen del sistema de coordenadas del vehículo sobre la recta, se calcula el punto objetivo en base a dos condiciones:

- Se encuentra a una distancia “look ahead distance” de la proyección del vehículo sobre la recta.
- Pertenece a la recta $Ax + By + C = 0$

Tal y como se puede apreciar en la Figura 20 existen dos puntos que cumplen las condiciones necesarias, uno de ellos se encuentra en la dirección de avance del vehículo (marcado en verde) y el otro en la dirección contraria (marcado en naranja). De cara a la implementación del algoritmo, será necesario por tanto calcular ambos puntos y discriminar cuál de ellos es el correcto en cada caso.

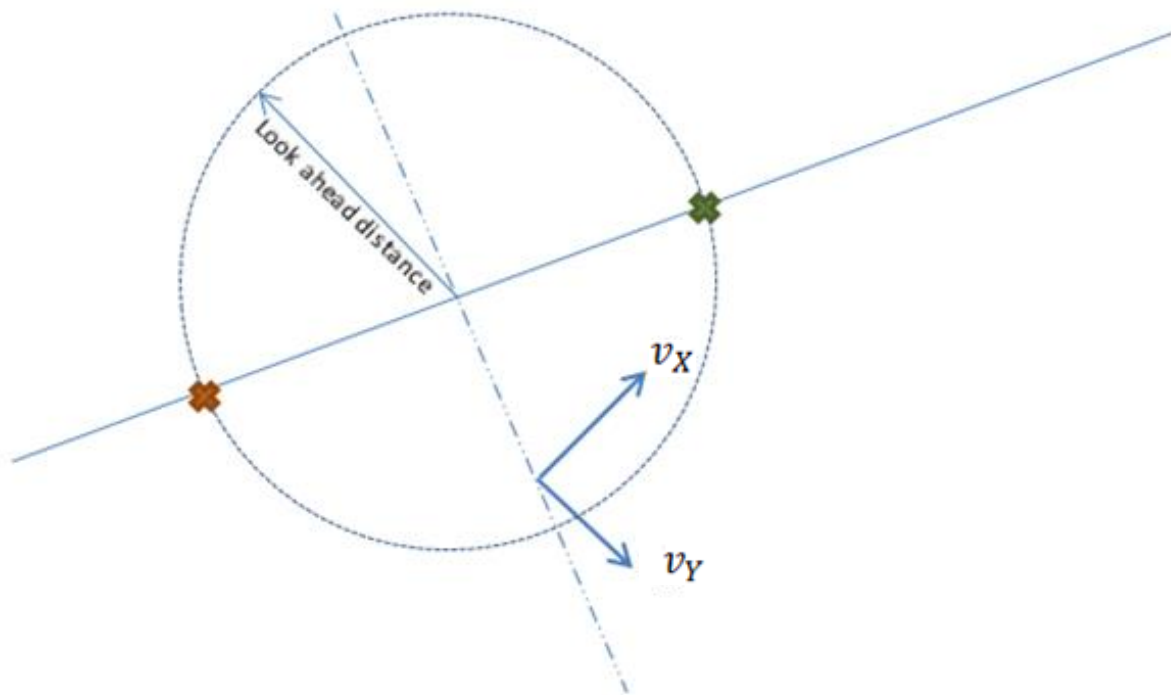


Figura 20: obtención del punto objetivo

Para obtener las coordenadas de los candidatos a punto objetivo, matemáticamente se plantea la intersección entre una recta y una circunferencia. La recta en cuestión será la recta de aproximación calculada, y la circunferencia estará definida por su centro y su radio, siendo éstos la proyección del vehículo sobre la recta, y la distancia “look ahead distance” respectivamente. Hay que tener en cuenta que la ecuación de la recta está referida al sistema de referencia global, por lo tanto, todas las coordenadas deben estar referidas a dicho sistema de referencia.

Esto se traduce en resolver el siguiente sistema de ecuaciones:

$$\begin{cases} Ax + By + C = 0 \\ R^2 = (x - P_x)^2 + (y - P_y)^2 \end{cases} \quad [3.10]$$

Dónde:

- (P_x, P_y) son las coordenadas de la proyección del origen de sistema de coordenadas del vehículo sobre la recta de aproximación, respecto al sistema de coordenadas global.
- R es el radio de la circunferencia, que como se ha dicho es el valor del parámetro “look ahead distance”.

Para resolver el sistema de ecuaciones 3.10, se despeja la primera ecuación de la siguiente manera:

$$y = -\frac{Ax + C}{B} \quad [3.11]$$

Teniendo en cuenta los resultados analizados en la sección 3.1.2 tal y como se resume en la Tabla 2 de dicha sección, es interesante observar que el parámetro B siempre vale "0" ó "-1". Lo cual resulta especialmente útil para resolver el sistema de ecuaciones 3.10, ya que se puede distinguir entre ambos casos:

Caso " $B = -1$ " (*rectas oblicuas u horizontales*)

La ecuación 3.11 se simplifica a:

$$y = Ax + C \quad [3.12]$$

Sustituyendo la ecuación 3.12 en el sistema de ecuaciones 3.10, se obtiene:

$$R^2 = (x - P_x)^2 + (Ax + C - P_y)^2 \quad [3.13]$$

Si se define:

$$t = C - P_y \quad [3.14]$$

Operando la ecuación 3.13, se obtiene una expresión de segundo grado para el valor de x :

$$x^2 \cdot (1 + A^2) + x \cdot (-2 \cdot P_x - 2 \cdot A \cdot t) + (t^2 + P_x^2 - R^2) = 0 \quad [3.15]$$

Definiendo:

$$a = 1 + A^2 \quad [3.16]$$

$$b = -2 \cdot P_x - 2 \cdot A \cdot t \quad [3.17]$$

$$c = t^2 + P_x^2 - R^2 \quad [3.18]$$

Se obtiene finalmente el valor de x aplicando la fórmula para la resolución de ecuaciones de segundo grado:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad [3.19]$$

Donde los valores de los parámetros a, b, c están definidos por las expresiones 3.16, 3.17 y 3.18 respectivamente.

Una vez obtenidos los dos posibles valores de " x " según la expresión 3.19, se calculan los correspondientes valores de " y " según la expresión 3.12.

Caso " $B = 0$ " (rectas verticales)

La ecuación 3.11 daría lugar a una indeterminación:

$$y = \lim_{B \rightarrow 0} -\frac{Ax + C}{B} \quad [3.20]$$

En la práctica se conoce que cuando el parámetro B es nulo, la ecuación de la recta corresponde con una recta vertical, cuya expresión es:

$$x = k \quad [3.21]$$

Sustituyendo la expresión de la recta definida por la ecuación 3.21 en el sistema de ecuaciones 3.10, el sistema quedará:

$$\begin{cases} x = k \\ R^2 = (x - P_x)^2 + (y - P_y)^2 \end{cases} \quad [3.22]$$

Resolviendo el sistema 3.22, se obtiene:

$$R^2 = (k - P_x)^2 + (y - P_y)^2 \quad [3.23]$$

Si se define:

$$t = k - P_x \quad [3.24]$$

Operando la ecuación 3.23, se obtiene una expresión de segundo grado para el valor de y :

$$y^2 + x \cdot (-2 \cdot P_y) + (t^2 + P_y^2 - R^2) = 0 \quad [3.25]$$

Definiendo:

$$a = 1 \quad [3.26]$$

$$b = -2 \cdot P_y \quad [3.27]$$

$$c = t^2 + P_y^2 - R^2 \quad [3.28]$$

Se obtiene finalmente el valor de y aplicando la fórmula para la resolución de ecuaciones de segundo grado:

$$y = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad [3.29]$$

Donde los valores de los parámetros a, b, c están definidos por las expresiones 3.26, 3.27 y 3.28 respectivamente.

Una vez obtenidos los dos posibles valores de " y " según la expresión 3.29, se construyen los dos puntos sabiendo que en ambos casos la componente " x " es igual a " k " según la ecuación 3.21

Como se dijo anteriormente, una vez calculados los dos candidatos a punto objetivo, es necesario determinar cuál de ellos es el correcto; para lo cual se establecen dos métodos.

Método 1

El primer método se basa en comparar los ángulos δ_1 y δ_2 formados entre los vectores que se representan en la Figura 21. El método presupone que el vehículo está correctamente orientado hacia la trayectoria deseada.

Dónde δ_1 es el ángulo formado entre el vector orientación actual del vehículo ("heading") determinado por el eje X del sistema de referencia del vehículo, y el vector P_1 cuyas componentes (x,y) son las coordenadas del punto objetivo 1 respecto al sistema de referencia del vehículo. La definición del ángulo δ_2 es análoga a la definición de δ_1 .

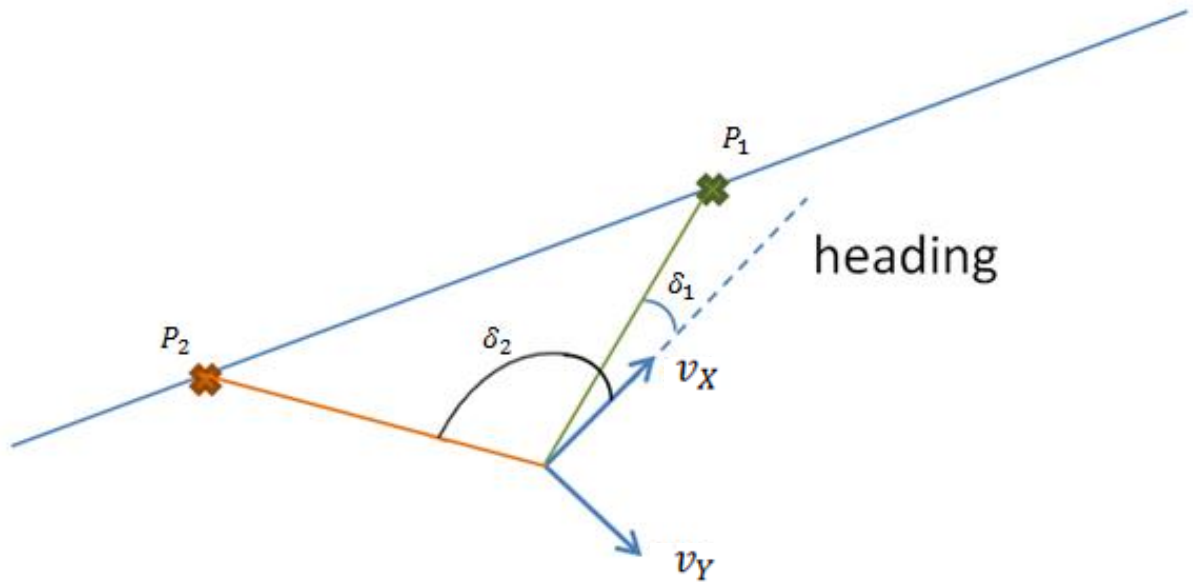


Figura 21: elección del punto objetivo, método 1

Matemáticamente, los ángulos δ_1 y δ_2 están definidos según las siguientes expresiones:

$$\delta_1 = \cos^{-1} \left(\frac{\bar{h} \cdot \overline{P_1}}{|\bar{h}| \cdot |P_1|} \right) = \cos^{-1} \left(\frac{h_x \cdot P_{1x} + h_y \cdot P_{1y}}{\sqrt{h_x^2 + h_y^2} \cdot \sqrt{P_{1x}^2 + P_{1y}^2}} \right) \quad [3.30]$$

$$\delta_2 = \cos^{-1} \left(\frac{\bar{h} \cdot \bar{P}_1}{|\bar{h}| \cdot |P_2|} \right) = \cos^{-1} \left(\frac{h_x \cdot P_{2x} + h_y \cdot P_{2y}}{\sqrt{h_x^2 + h_y^2} \cdot \sqrt{P_{2x}^2 + P_{2y}^2}} \right) \quad [3.31]$$

Ecuaciones 3.30 y 3.31 que pueden simplificarse, teniendo en cuenta que la componente h_y del vector orientación (“heading”) es nula y la componente h_x es 1, ya que las coordenadas de los vectores están expresadas en el sistema de referencia del vehículo, tal y como se puede ver en la Figura 21. Decir que la componente $h_x = 1$ es hacer el vector orientación unitario.

$$\delta_1 = \cos^{-1} \left(\frac{\bar{h} \cdot \bar{P}_1}{|\bar{h}| \cdot |P_1|} \right) = \cos^{-1} \left(\frac{P_{1x}}{\sqrt{P_{1x}^2 + P_{1y}^2}} \right) \quad [3.32]$$

$$\delta_2 = \cos^{-1} \left(\frac{\bar{h} \cdot \bar{P}_1}{|\bar{h}| \cdot |P_2|} \right) = \cos^{-1} \left(\frac{P_{2x}}{\sqrt{P_{2x}^2 + P_{2y}^2}} \right) \quad [3.33]$$

Una vez calculados los dos ángulos, para decidir cuál de los puntos calculados anteriormente es el punto objetivo correcto, es suficiente con hacer una comparación entre ambos valores. Tal y como se aprecia en la Figura 21, suponiendo que el vehículo está correctamente orientado hacia la trayectoria, la solución correcta será el punto cuyo ángulo asociado sea menor.

Método 2

El segundo método se basa en utilizar los waypoints que definen la recta actual, para decidir cuál de los dos candidatos a punto objetivo es el correcto. Al igual que el método 1, se presupone que el vehículo está correctamente orientado hacia la trayectoria deseada.

La idea consiste en calcular la distancia punto a punto, o distancia euclídea, entre uno de los waypoints de la recta actual y los dos candidatos a punto objetivo. En la Figura 22 se representa gráficamente lo comentado hasta ahora. Los waypoints se representan en azul y los dos posibles puntos objetivo se representan en verde y naranja. En color púrpura se destaca el cuarto waypoint, que será aquel que se utilice para calcular ambas distancias. La razón de elegir éste waypoint y no otro es sencillamente que se encuentra aproximadamente en la mitad entre los ocho waypoints utilizados, aunque cualquier otro waypoint exceptuando el primero, sería igualmente válido para el método. Se recuerda que el primer waypoint es un punto del camino deseado que el vehículo ya ha sobrepasado, por lo que no sería válido.

La distancia euclídea entre los puntos Wp_4 y P_1 vendrá dada por la expresión:

$$dist_1 = \sqrt{(Wp_{4x} - P_{1x})^2 + (Wp_{4y} - P_{1y})^2} \quad [3.34]$$

Análogamente, La distancia euclídea entre los puntos Wp_4 y P_2 vendrá dada por la expresión:

$$dist_2 = \sqrt{(Wp_{4x} - P_{2x})^2 + (Wp_{4y} - P_{2y})^2} \quad [3.35]$$

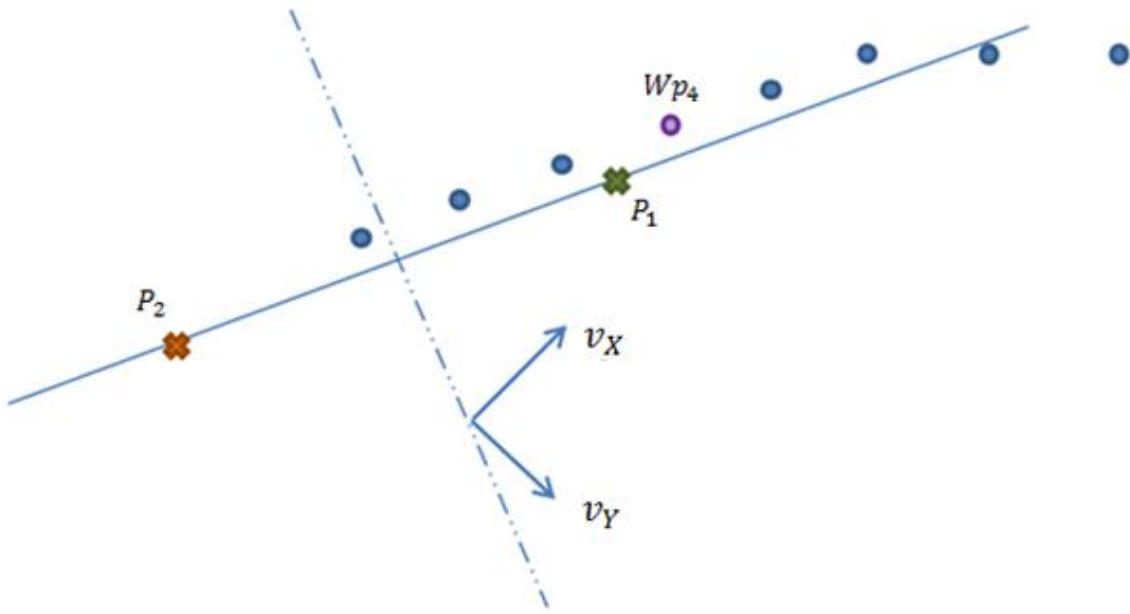


Figura 22: elección del punto objetivo, método 2

Tal y como se puede apreciar en la Figura 22, suponiendo que la orientación del vehículo es la adecuada, el punto objetivo correcto será aquel cuya distancia al waypoint Wp_4 sea menor.

Ambos métodos parten de la premisa de que el vehículo este orientado en la dirección de avance. No obstante, a diferencia del primer modelo, este segundo método no es sensible a la orientación actual del vehículo, ya que en ningún momento se utiliza el vector “heading” en los cálculos. Lo cual puede considerarse inclusive una ventaja, ya que si hubiese algún error en la actualización de la orientación, se seguiría escogiendo el punto objetivo correcto. Es posible realizar esta afirmación puesto que los waypoints vienen dados en orden e indican la dirección en que se desea comandar el vehículo.

3.2.3 Cálculo del ángulo de dirección

Una vez calculado el punto objetivo, el siguiente paso del algoritmo Pure Pursuit es calcular el ángulo de dirección necesario para que el vehículo alcance dicho punto. Los problemas geométricos derivados de éste cálculo se pueden resolver tanto en el sistema de referencia global, como en el sistema de referencia del vehículo. No obstante, las condiciones finales son más sencillas si se resuelven en el sistema del vehículo.

Cómo se describió en la sección 2.2.2, el problema se resuelve calculando el radio de curvatura, definido por la expresión:

$$R = \frac{Goal}{2 \cdot \sin(\alpha)} \quad [3.36]$$

Dónde:

- “ α ” es el ángulo que forman los vectores “heading” y “Goal”, ambos relativos al sistema de referencia del vehículo.
- “Goal” es la distancia euclídea desde el origen del sistema de referencia del vehículo hasta el punto objetivo.
- “R” es el radio de la curvatura que describe el vehículo.

Comparando la ecuación 3.36 con la ecuación 2.7, se aprecia el cambio en el denominador del parámetro look “ahead distance”, por la distancia “Goal”.

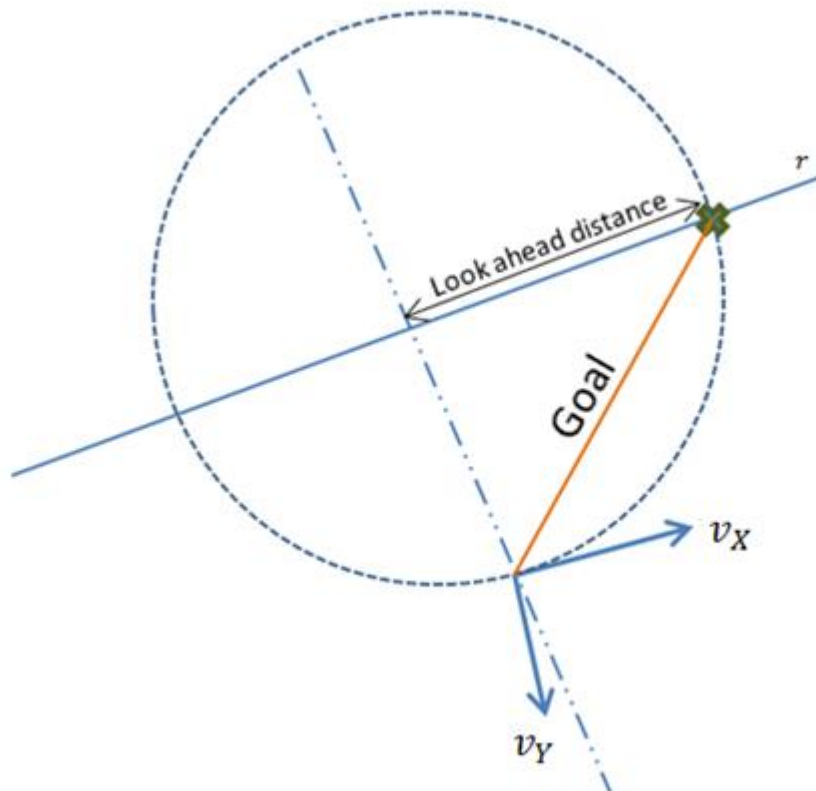


Figura 23: diferencia entre look ahead distance y distancia Goal

El cambio de una distancia por otra se debe al hecho de calcular la proyección del vehículo sobre la recta de aproximación como paso previo a la obtención del punto objetivo. Si no se hiciera ese proceso, el denominador de la ecuación 3.36 sería el parámetro “look ahead distance”. El motivo de calcular la proyección del vehículo es asegurar que siempre exista una solución para el sistema de ecuaciones 3.10, cuya resolución se explicó la sección 3.2 2.

En la Figura 23, se puede apreciar la diferencia entre estas dos distancias, donde:

- Los subíndices “ v_x , v_y ” representan el sistema de referencia del vehículo.
- El punto objetivo está representado con una cruz verde
- La circunferencia discontinua representa la trayectoria descrita por el vehículo
- la intersección entre la recta “ r ” y la línea perpendicular a ella que pasa por el origen del sistema de referencia del vehículo, denota la proyección del vehículo sobre la recta.

Al aplicar el método de la proyección, los valores de “look ahead distance” y “Goal” sólo coinciden cuando el vehículo describe una trayectoria totalmente recta.

Una vez explicado el cambio en la ecuación 3.36, se analiza en detalle la obtención del ángulo de dirección. En la Figura 24 se ilustra el problema geométrico a resolver, donde las incógnitas son “ R ” y “ (C_x, C_y) ”, que son el radio y las coordenadas del centro de la circunferencia respecto al sistema de referencia del vehículo, respectivamente. Se presentan dos soluciones, la primera trabajando con el sistema de referencia del vehículo, y la segunda trabajando en el sistema de referencia global.

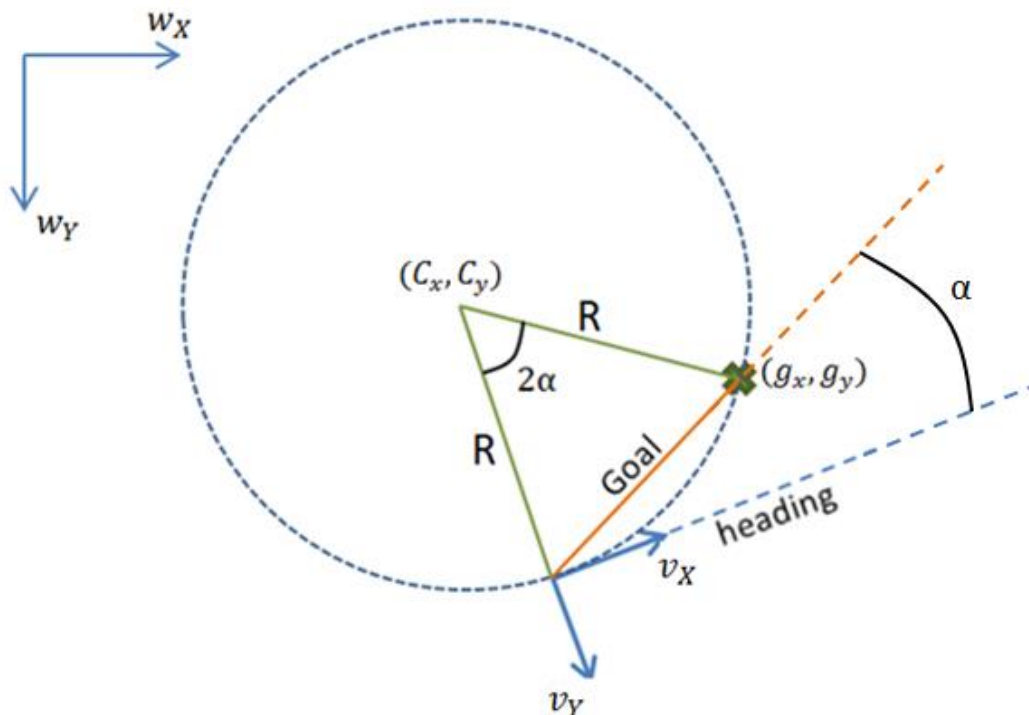


Figura 24: Parámetros Pure Pursuit

Método 1: solución en el sistema de referencia del vehículo

Para calcular el radio de curvatura de la Figura 24 a partir de la ecuación 3.36, es necesario obtener en primer lugar el ángulo “ α ”, que cómo se analizó en la sección 2.2.2, es el ángulo que forman los vectores “heading” y “goal”. Dicho ángulo se calcula según la expresión:

$$\alpha = \cos^{-1} \left(\frac{\bar{h} \cdot \bar{g}}{|\bar{h}| \cdot |\bar{g}|} \right) \quad [3.37]$$

Dónde:

- \bar{h} es el vector “heading” que indica la orientación actual del vehículo. Expresado en el sistema de referencia de referencia del vehículo sus componentes siempre son (1,0).
- \bar{g} es el vector “goal” que indica la posición del punto objetivo en el sistema de referencia del vehículo.

No obstante, el sistema de referencia del vehículo es móvil, es decir, tanto su posición como su orientación son variables. Este hecho implica que para resolver la ecuación 3.37 hay que obtener las coordenadas del vector “goal” en el sistema de referencia del vehículo, lo cual se traduce en aplicar una matriz de rotación para pasar de un sistema de referencia a otro. Es decir, aplicar la transformación:

$$\begin{pmatrix} v_X \\ v_Y \end{pmatrix} = \begin{pmatrix} \cos \beta & \sin \beta \\ -\sin \beta & \cos \beta \end{pmatrix} \begin{pmatrix} w_X \\ w_Y \end{pmatrix} \quad [3.38]$$

Dónde (v_X, v_Y) son las coordenadas de cualquier vector en el sistema de referencia del vehículo, (w_X, w_Y) son las coordenadas de ese mismo vector en el sistema de referencia global, la matriz de dimensión 2x2 es la matriz de rotación y la amplitud β es el ángulo que forman los ejes, w_Y e v_Y del sistema de referencia global y del vehículo respectivamente, ángulo que coincide con la expresión:

$$\beta = \frac{\pi}{2} + \theta \quad [3.39]$$

Siendo θ en ángulo de orientación actual del vehículo, que es el ángulo formado entre los ejes w_Y e v_X del sistema de referencia global y del vehículo respectivamente, cuya representación se puede ver en la Figura 25.

El vector (w_X, w_Y) sobre el que hay que aplicar la ecuación 3.38 es el vector relativo “goal” expresado en el sistema de referencia global, cuya expresión es:

$$\bar{g} = \bar{G} - \bar{V} \quad [3.40]$$

Dónde los vectores \bar{V} y \bar{G} son el vector que indica la posición del vehículo en el sistema de referencia global, y el vector que indica la posición del punto objetivo en el sistema de referencia global respectivamente. Ambos vectores se representan en la Figura 25.

De esta forma la ecuación 3.37 se resolvería finalmente obteniendo el valor de ángulo “ α ”, según:

$$\alpha = \cos^{-1} \left(\frac{g_x^v}{\sqrt{(g_x^v)^2 + (g_y^v)^2}} \right) \quad [3.41]$$

Dónde (g_x^v, g_y^v) denotan las coordenadas del vector “goal” expresadas en el sistema de referencia del vehículo.

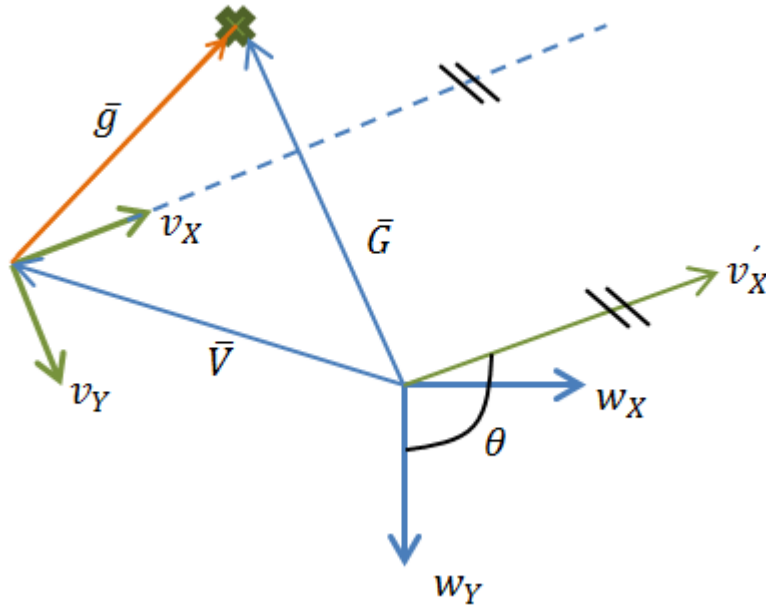


Figura 25: sistemas de referencia y vectores implicados en el cálculo del ángulo α

Una vez obtenido el ángulo “ α ”, a través de la expresión 3.41, se procede a calcular el radio de curvatura “ R ” de la Figura 24. Para lo cual se hace uso de la expresión 3.36, definida al principio de ésta sección.

No obstante, de cara a la implementación del código hay que tener en cuenta que la expresión 3.36 se convierte en una indeterminación cuando el ángulo “ α ” tiende a cero:

$$R = \lim_{\alpha \rightarrow 0} \frac{Goal}{2 \cdot \sin(\alpha)} \quad [3.42]$$

Por consiguiente, el algoritmo implementado debe analizar dicho caso antes de intentar resolver la expresión 3.36. Observando la Figura 24, es posible deducir que cuando $\alpha = 0$ el vehículo se encuentra perfectamente orientado hacia el punto objetivo, por lo que la trayectoria que debe realizar es una línea recta. Por tanto, en caso en que $\alpha = 0$ el ángulo de dirección necesario también será nulo, $\delta = 0$. También en este punto se consideran las restricciones físicas del vehículo, el mínimo ángulo de giro que permite distinguir el sistema de

dirección instalado en el vehículo es de 1° . Por lo tanto si el ángulo “ α ” calculado, es inferior a éste valor, se considera que el ángulo de dirección es cero, es decir, $\delta = 0$.

Se aprovecha éste caso para aclarar lo que ya resulta evidente, y es que el ángulo “ α ” de la Figura 24 y el ángulo de dirección del vehículo “ δ ” no son iguales. El ángulo de dirección del vehículo se obtiene en base al modelo de Ackerman según la expresión 2.4 que se analizó en la sección 2.1.2, utilizando como radio el parámetro “ R ” de la Figura 24.

Volviendo a la expresión 3.36, en caso de que el ángulo “ α ” no sea nulo, se procede a calcular el valor de “ R ”. Sustituyendo los resultados obtenidos, la expresión 3.36 será:

$$R = \frac{Goal}{2 \cdot \sin(\alpha)} = \frac{\sqrt{g_x^2 + g_y^2}}{2 \cdot \sin \alpha} \quad [3.43]$$

Dónde:

· (g_x, g_y) son las coordenadas del punto objetivo respecto al sistema de referencia del vehículo.

Una vez obtenido el valor de “ R ” se calcula el ángulo de dirección “ δ ” mediante la expresión 2.4, dónde “ L ” es la distancia entre los ejes delantero y trasero del vehículo, que en el caso de la aplicación son 1.66 metros:

$$\delta = \tan^{-1}\left(\frac{L}{R}\right) = \tan^{-1}\left(\frac{1.66}{R}\right) \quad [3.44]$$

Hasta aquí, se ha explicado cómo calcular el ángulo de dirección del vehículo, no obstante la expresión 3.44 siempre dará como resultado un valor positivo, ya que tanto “ L ” como “ R ” son distancias físicas y por tanto valores positivos, y la función arcotangente de un número positivo siempre es positiva. Hecho que nos lleva a discutir sobre el signo del ángulo de dirección y su significado.

Siempre que se ha hablado de los sistema de referencia utilizados, se consideran sólo dos ejes (X,Y). Esto es así porque estamos tratando con un vehículo autónomo terrestre y por tanto el movimiento se produce en el plano. No obstante los ángulos definidos y los giros son en verdad movimientos de rotación sobre el eje Z perpendicular al plano. En base a la Figura 24, siguiendo la regla de la mano derecha ambos sistemas de referencia, el global y el del vehículo, tienen su eje Z en la dirección perpendicular al papel “introduciéndose” en éste. Es decir, los giros en sentido horario son positivos; y los giros en sentido anti-horario son negativos. Por consiguiente, si el ángulo de dirección del vehículo “ δ ” es positivo el vehículo realizará un giro horario, es decir girará a “su derecha”; si por el contrario “ δ ” es negativo, el vehículo realizará un giro anti-horario, es decir girará a “su izquierda”.

Dado que la expresión 3.44 siempre dará como resultado un ángulo positivo, será necesario averiguar de alguna forma si el giro que debe realizar el vehículo es anti-horario u horario, a fin de asignar al ángulo de dirección su signo correspondiente en cada caso. Para

ello, se propone a continuación un método que precisa calcular el centro de rotación definido como (C_x, C_y) en la Figura 24.

Para calcular el centro de rotación se plantea un sistema de ecuaciones entre una recta y una circunferencia, cuya intersección será dicho centro. Efectivamente, al utilizar una ecuación de grado 2 en el sistema de ecuaciones, éste tendrá dos posibles soluciones, que corresponden con los dos posibles centros de las circunferencias que podría describir el vehículo para llegar hasta el punto objetivo calculado. No obstante sólo uno de estos movimientos es posible tal y cómo se discutirá posteriormente.

En la Figura 26 se puede ver el problema geométrico descrito. Las circunferencias discontinuas de color verde y rojo muestran las posibles trayectorias del vehículo desde su posición actual (marcada por el sistema de referencia del vehículo) hasta el punto objetivo (marcado con una cruz verde). El sistema de ecuaciones planteado es para calcular la intersección entre la recta “s”, que es perpendicular al vector “Goal” y pasa por el punto medio del segmento homónimo, y la circunferencia de color negro cuyo centro es el origen del sistema de referencia del vehículo y cuyo radio es el parámetro “R” obtenido mediante la expresión 3.43.

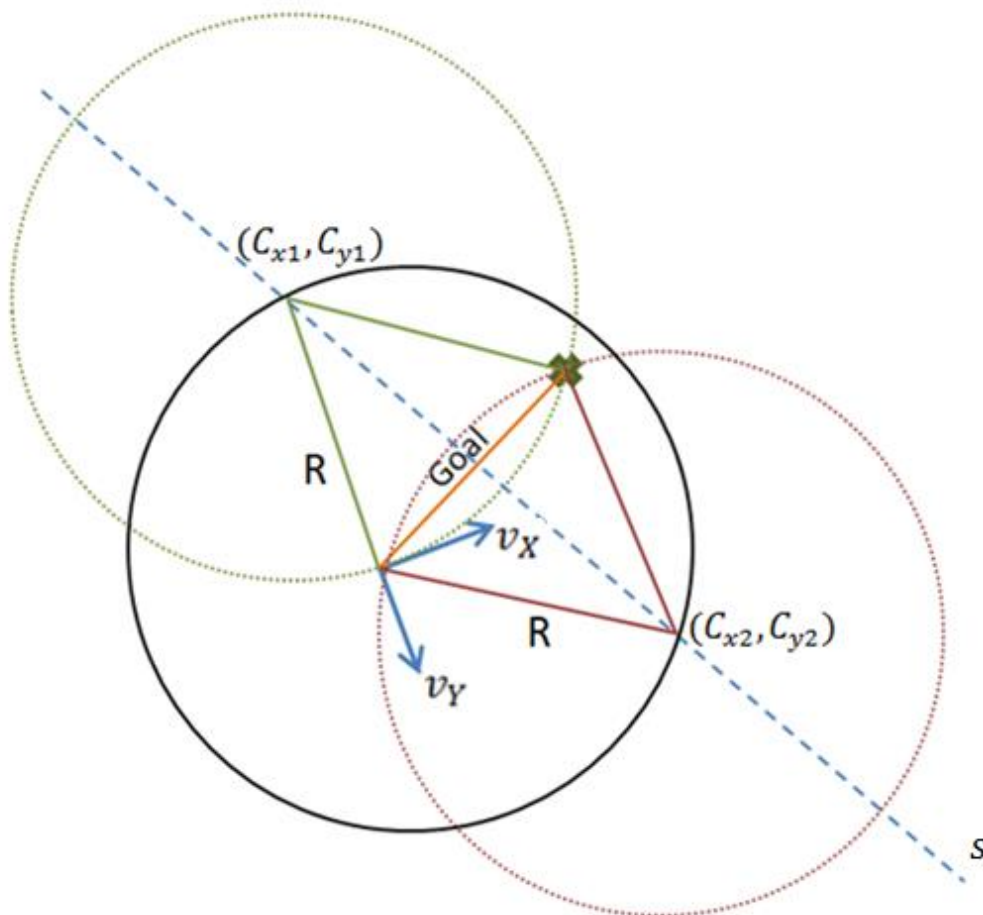


Figura 26: cálculo del centro de giro

El sistema de ecuaciones a resolver para calcular los centros (C_{x1}, C_{y1}) y (C_{x2}, C_{y2}) será por consiguiente:

$$\begin{cases} y = \frac{-g_x}{g_y} \cdot \left(x - \frac{g_x}{2}\right) + \frac{g_y}{2} \\ x^2 + y^2 = R^2 \end{cases} \quad [3.45]$$

Dónde:

(g_x, g_y) son las coordenadas del punto objetivo en el sistema de referencia del vehículo.

Para resolver el sistema de ecuaciones 3.45 en primer lugar hay que fijar la atención sobre la primera ecuación, que corresponde a la recta “s” de la Figura 26. Se aprecia que si la componente g_y del vector “goal” tiende a cero, se da una indeterminación, por lo que hay que considerar dicho caso a la hora de implementar el algoritmo.

Caso 1, $g_y \rightarrow 0$

Si g_y tiende a cero, nos encontramos con que la recta “s” es vertical en el sistema de referencia del vehículo, lo cual implica que el vector “goal” es horizontal en dicho sistema de referencia, ya que la recta “s” es perpendicular a dicho vector. Es decir, atendiendo a la Figura 26, el vector “goal” estaría completamente alineado con el eje X del sistema de referencia del vehículo. Lo cual implica que la trayectoria que debería seguir el vehículo para alcanzar el punto objetivo es una línea recta, por tanto el ángulo de dirección es nulo, y no es necesario resolver el sistema de ecuaciones 3.45 ya que el vehículo no tiene que hacer ningún giro y no hay que calcular el centro de rotación de dicho movimiento.

Caso 2, $g_y \neq 0$

En este caso, el ángulo de dirección será distinto de cero, es decir, el vehículo si describirá un movimiento circular, y por tanto será necesario resolver el sistema de ecuaciones 3.45.

Sustituyendo la primera ecuación en la segunda y operando la expresión, se obtiene:

$$x^2 + \left(\frac{-g_x}{g_y} \left(x - \frac{g_x}{2}\right) + \frac{g_y}{2}\right)^2 = R^2 \quad [3.46]$$

$$x^2 + \left(\frac{-g_x}{g_y} \cdot x + \frac{g_x^2}{2g_y} + \frac{g_y}{2}\right)^2 = R^2$$

$$x^2 + \left(\frac{g_x^2 + g_y^2}{2g_y} - \frac{g_x}{g_y} \cdot x\right)^2 = R^2$$

$$x^2 + \left(\frac{g_x^2 + g_y^2}{2g_y}\right)^2 + \frac{g_x^2}{g_y^2} \cdot x^2 - x \cdot \frac{g_x}{g_y} \cdot \frac{g_x^2 + g_y^2}{g_y} = R^2 \quad [3.47]$$

Si se define:

$$t = \frac{g_x^2 + g_y^2}{g_y} \quad [3.48]$$

La expresión 3.51 queda finalmente:

$$x^2 \cdot \left(1 + \frac{g_x^2}{g_y^2}\right) + x \cdot \left(\frac{-g_x}{g_y} \cdot t\right) + \left(\frac{t^2}{4g_y} - R^2\right) = 0 \quad [3.49]$$

Definiendo:

$$a = 1 + \frac{g_x^2}{g_y^2} \quad [3.50]$$

$$b = \frac{-g_x}{g_y} \cdot t \quad [3.51]$$

$$c = \frac{t^2}{4g_y} - R^2 \quad [3.52]$$

Se resuelve finalmente la componente “x” según:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad [3.53]$$

Una vez obtenidos los dos valores de “x” se obtienen los correspondientes valores de “y” utilizando la primera ecuación del sistema 3.45.

Puede ocurrir que la expresión 3.53 no tenga solución, en caso de que el radicando sea negativo. Físicamente eso implica que el vehículo no puede alcanzar el punto objetivo calculado y por tanto es necesario obtener uno nuevo aumentando el valor del parámetro “look ahead distance” o modificando la trayectoria a la que se aproxima el vehículo.

Con ambos centros calculados, falta discernir cuál de ellos es el correcto. Fijando la atención una vez más en la Figura 26, se puede comprobar que a pesar de existir dos posibles trayectorias, el vehículo solamente puede realizar la trayectoria descrita por la circunferencia verde. Esto es así dada a su orientación actual, ya que debido a la configuración Ackerman, en cada instante el movimiento está restringido en la dirección del eje X del sistema de referencia del vehículo.

Teniendo esto en cuenta, se puede deducir de la Figura 26, que el centro de giro correcto será aquel cuyo radio (desde el origen del sistema de referencia del vehículo) forme un ángulo mayor con el eje X del sistema de referencia del vehículo, es decir con su orientación actual. Matemáticamente dicho ángulo debería ser 90°, no obstante en la práctica el valor obtenido puede tener muchos decimales, y por tanto es recomendable hacer una comparación entre ambos eligiendo el mayor.

En la Figura 27 se pueden apreciar con mayor claridad los ángulos descritos en el párrafo anterior, β_1 y β_2 . Observando dicha figura, también se puede apreciar que, ya que los centros están referidos al sistema de coordenadas del vehículo, no es siquiera necesario calcular los ángulos. Es posible elegir el centro correcto como aquel cuya componente X sea menor, (teóricamente cero). No obstante habría que hacer la comparación en valor absoluto para no tener en cuenta el signo negativo.

Una vez calculados los posibles centros de giro, y elegido cuál de ellos es el correcto, falta discernir el signo del ángulo de dirección. Para lo cual se analiza la posición del centro de giro respecto al sistema de referencia del vehículo.

Si la componente “y” del centro de giro es negativa, el giro que debe hacer el vehículo es en sentido anti-horario. Si por el contrario la componente “y” del centro de giro es positiva, el giro es en sentido horario.

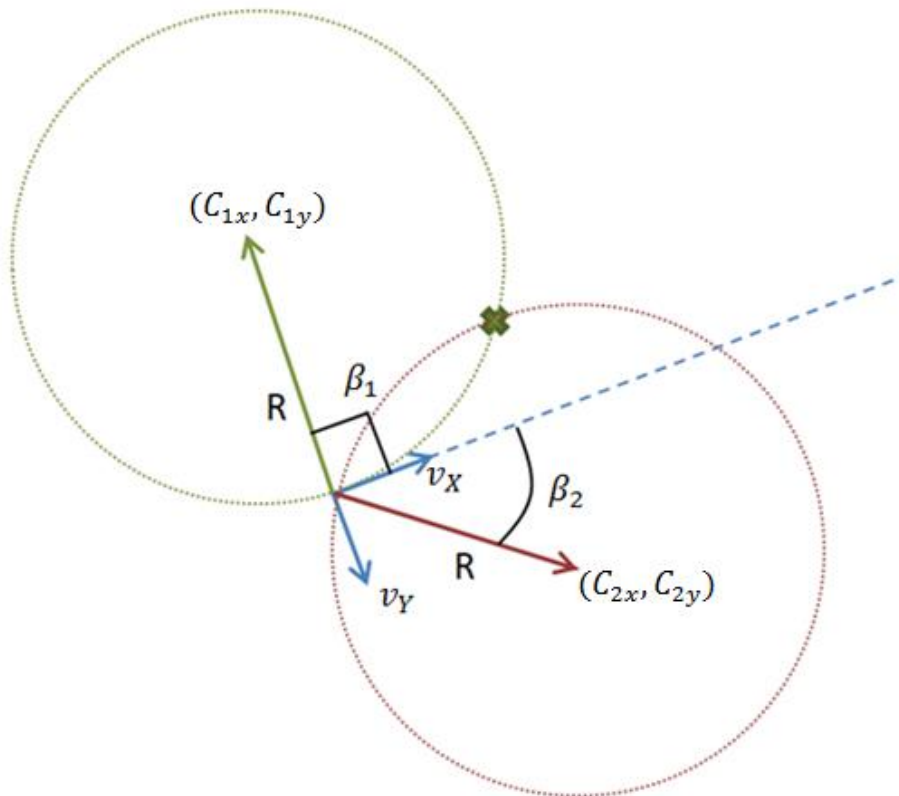


Figura 27: elección del centro de giro correcto, método 1

En resumen:

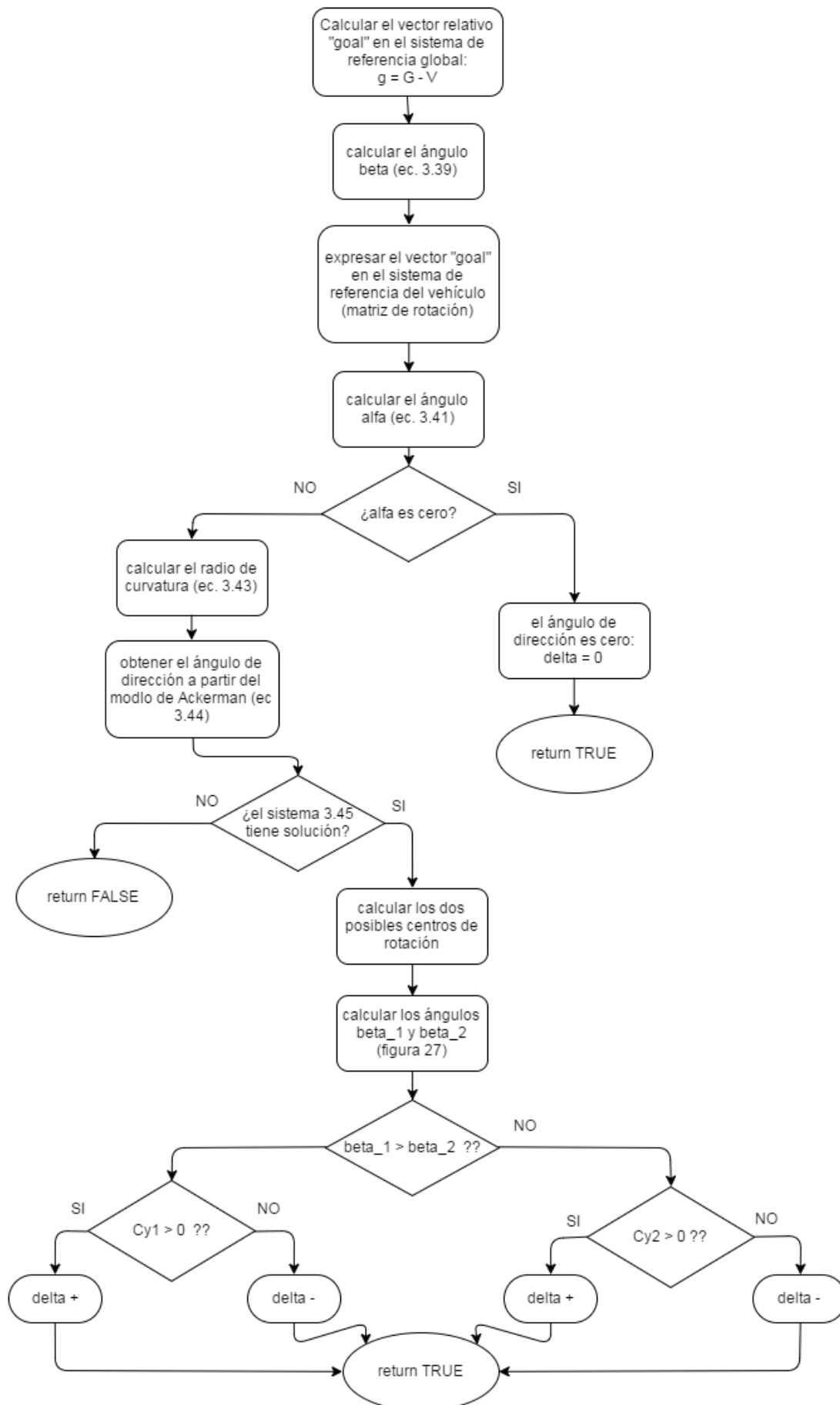
$$\cdot \text{ si } C_y > 0 \rightarrow \delta > 0$$

$$\cdot \text{ si } C_y < 0 \rightarrow \delta < 0$$

Dónde “ δ ” es el resultado de la expresión 3.44

Con lo que finalmente se ha alcanzado el objetivo del algoritmo Pure Pursuit y ya es posible comandar el vehículo hacia el punto objetivo elegido originalmente. A continuación se

representa en un flujograma la ejecución de una función de C++ basada en el método 1.



Fujograma 1: cálculo del ángulo de dirección método_1 52

Método 2: solución en el sistema de referencia global

Si se trabaja en el sistema de referencia global, se evita tener que hacer uso de las transformaciones lineales entre sistemas de ecuaciones. Las ecuaciones planteadas son más sencillas de obtener pero las condiciones finales para elegir el signo del ángulo de dirección son más complejas. Para calcular el centro de rotación se hace uso del sistema de referencia del vehículo, pero considerando que éste no ha rotado respecto al sistema de referencia global, únicamente se traslada, es decir altera su posición. Esto permite que se puedan hacer sumas vectoriales directamente entre vectores expresados en cualquiera de los dos sistemas de referencia, ya que la relación entre ambos es simplemente una transformación lineal de traslación, que corresponde con el vector de posición del vehículo (vector \bar{V} de la Figura 25), o lo que es igual, el vector que une los orígenes de ambos sistemas de referencia.

Dado que las coordenadas del vector relativo “goal” en el sistema de referencia global son conocidas, y las componentes del vector “heading” en el sistema de referencia global también lo son, es mucho más sencillo calcular el ángulo “ α ” operando con dichos vectores expresados en el sistema de referencia global. De ésta manera, la amplitud “ α ” se calcularía según la expresión 3.37 dónde:

$$\bar{h} = (h_x, h_y) = (-\sin \theta, \cos \theta) \quad [3.54]$$

$$\bar{g} = \bar{G} - \bar{V} \quad [3.55]$$

Siendo el ángulo “ θ ” y los vectores \bar{V} y \bar{G} los definidos en la Figura 25.

De esta manera, la ecuación 3.37 se resolvería obteniendo el valor del ángulo “ α ” según:

$$\alpha = \cos^{-1} \left(\frac{h_x^w \cdot g_x^w + h_y^w \cdot g_y^w}{\sqrt{(h_x^w)^2 + (h_y^w)^2} \cdot \sqrt{(g_x^w)^2 + (g_y^w)^2}} \right) \quad [3.56]$$

Expresión 3.56 que resulta a priori más compleja que la expresión 3.41, pero que sin embargo no lo es, ya que las componentes de los vectores referenciados al sistema global son de obtención directa a través de las ecuaciones 3.54 y 3.55.

Al igual que ocurría en el método 1, si el ángulo “ α ” calculado es cero, implica que el ángulo de dirección “ δ ” también es nulo.

Una vez calculado el ángulo “ α ” se obtiene “ R ” según la expresión 3.36. Al igual que se razonó anteriormente, para evitar la indeterminación 3.42, hay que considerar el caso en que el ángulo “ α ” calculado sea cero, lo cual implica que el vehículo debe seguir una trayectoria rectilínea y por tanto el ángulo de dirección a aplicar “ δ ” es nulo.

$$R = \frac{Goal}{2 \cdot \sin(\alpha)} = \frac{\sqrt{(V_x - G_x)^2 + (V_y - G_y)^2}}{2 \cdot \sin \alpha} \quad [3.57]$$

Dónde:

- (V_x, V_y) son las coordenadas que indican el origen del sistema de referencia del vehículo, expresadas en el sistema de referencia global.

- (G_x, G_y) son las coordenadas del punto objetivo en el sistema de referencia global.

Una vez obtenido el valor de “ R ” se calcula el ángulo de dirección “ δ ” mediante la expresión 2.4, donde “ L ” es la distancia entre los ejes delantero y trasero del vehículo, que en el caso de la aplicación son 1.66 metros:

$$\delta = \tan^{-1}\left(\frac{L}{R}\right) = \tan^{-1}\left(\frac{1.66}{R}\right) \quad [3.58]$$

Al igual que sucedía en el método 1, hay que calcular el centro de rotación para asignar un signo al ángulo de dirección. El problema geométrico a resolver es exactamente el mismo que el descrito en la Figura 26. El sistema de ecuaciones planteado para calcular los centros (C_{x1}, C_{y1}) y (C_{x2}, C_{y2}) será por consiguiente:

$$\begin{cases} y = \frac{-g_x}{g_y} \cdot \left(x - \frac{g_x}{2}\right) + \frac{g_y}{2} \\ x^2 + y^2 = R^2 \end{cases} \quad [3.59]$$

Dónde:

- (g_x, g_y) son las coordenadas del punto objetivo en el sistema de referencia global, trasladado al sistema de referencia del vehículo. Es decir, se resuelve en el sistema de referencia del vehículo, pero éste no ha rotado con respecto al sistema de referencia global, únicamente se ha trasladado. Ese detalle hace que la ecuación de la circunferencia esté centrada en el origen, y al mismo tiempo se evita hacer uso de las matrices de rotación. Por tanto la solución es más sencilla de obtener.

Para resolver el sistema de ecuaciones 3.57 en primer lugar hay que fijar la atención sobre la primera ecuación, que corresponde a la recta “ s ” de la Figura 26. Se aprecia que si la componente g_y del vector “goal” tiende a cero, se da una indeterminación, por lo que hay que considerar dicho caso a la hora de implementar el algoritmo.

Caso 1, $g_y \rightarrow 0$

En este caso, cómo las ecuaciones del sistema 3.59 están planteadas en el sistema de referencia del vehículo pero éste no ha rotado con respecto al sistema de referencia global, el hecho de que g_y sea cero, significa simplemente que la recta “ s ” de la Figura 26 es vertical.

Por tanto, se plantea el sistema de ecuaciones 3.59 de la siguiente manera:

$$\begin{cases} x = \frac{g_x}{2} \\ x^2 + y^2 = R^2 \end{cases} \quad [3.60]$$

Por lo que resolviendo el sistema para la componente “y”, se tiene una ecuación de segundo grado incompleta, cuya solución es:

$$y = \sqrt{R^2 - \left(\frac{g_x}{2}\right)^2} \quad [3.61]$$

Expresión 3.61 que en caso de que el radicando sea negativo no tiene solución, lo cual implica que el punto objetivo calculado no es alcanzable, y se debe repetir el cálculo del punto objetivo aumentando el valor del parámetro “look ahead distance” o modificando la trayectoria a la que se aproxima el vehículo.

Caso 2, $g_y \neq 0$

En este caso, la recta “s” de la Figura 26 es horizontal u oblicua, y la solución del sistema de ecuaciones 3.49 es idéntica a la desarrollada en las ecuaciones 3.46 a 3.53.

Con ambos centros calculados, se establece el siguiente criterio para determinar cuál de ellos es el centro de giro correcto, y establecer así el signo del ángulo de dirección. De manera muy similar al criterio del método 1, se decide cual es el centro de giro correcto, para lo cual se calculan los ángulos β_1 y β_2 de la Figura 28. Se comparan entre sí y se elige el mayor de ellos cuyo centro asociado será el centro de giro correcto.

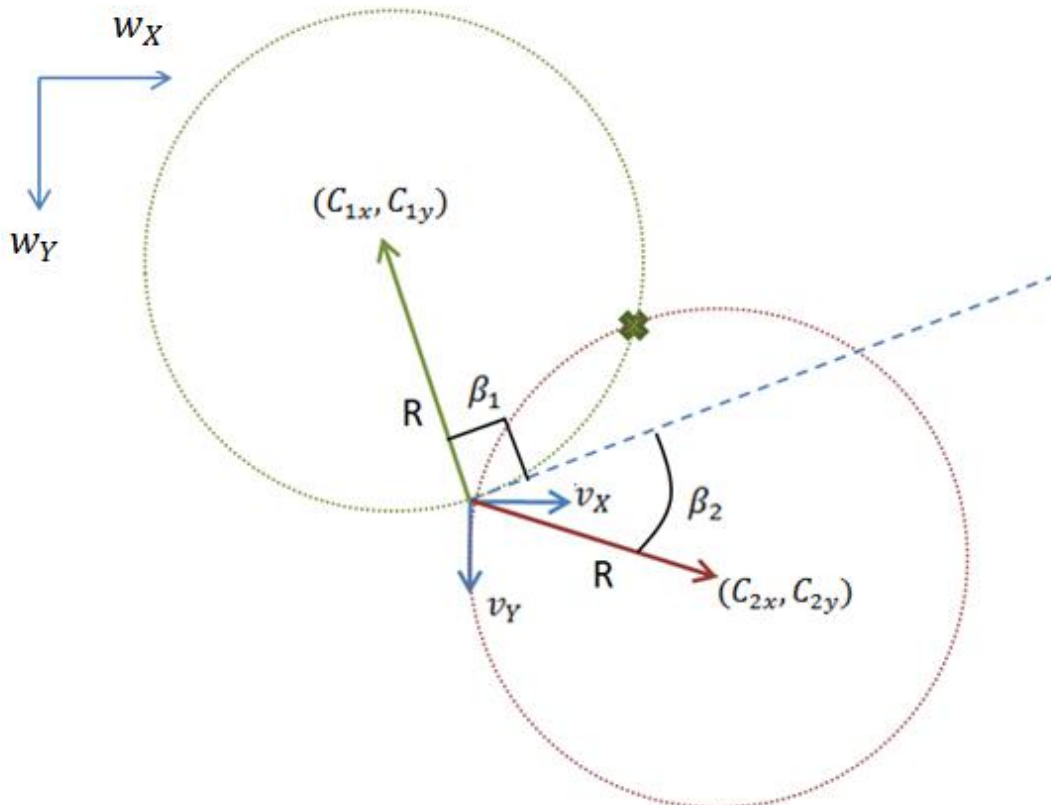


Figura 28 : ángulos Beta1 y Beta2, método 2

Una vez determinado el centro de giro correcto, se comprueba el signo de las componentes del vector “goal”, para determinar el signo del ángulo de dirección. Se distingue entre los cuatro cuadrantes del plano XY estableciendo el siguiente criterio, siendo C_x, C_y las coordenadas del centro de giro correcto, y g_x, g_y las coordenadas del vector goal. En la Figura 29 se representan los vectores implicados y la numeración de los cuadrantes.

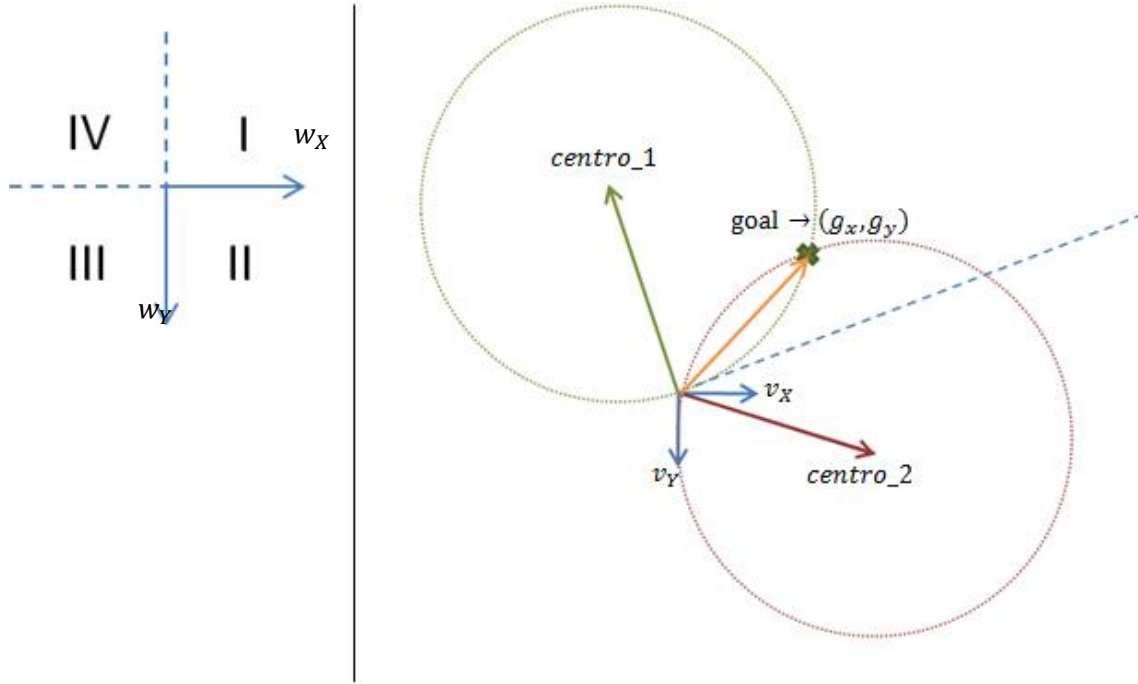


Figura 29: criterio para el signo del ángulo de dirección método 2.

· Si el vector goal está en el primer cuadrante: $g_x > 0$ y $g_y < 0$

· Si $C_x - \frac{g_x}{2} > 0 \rightarrow \delta > 0$

· Si $C_x - \frac{g_x}{2} < 0 \rightarrow \delta < 0$

· Si el vector goal está en el segundo cuadrante: $g_x < 0$ y $g_y > 0$

· Si $C_x - \frac{g_x}{2} > 0 \rightarrow \delta < 0$

· Si $C_x - \frac{g_x}{2} < 0 \rightarrow \delta > 0$

· Si el vector goal está en el tercer cuadrante: $g_x < 0$ y $g_y < 0$

· Si $C_x - \frac{g_x}{2} > 0 \rightarrow \delta < 0$

· Si $C_x - \frac{g_x}{2} < 0 \rightarrow \delta > 0$

· Si el vector goal está en el cuarto cuadrante: $g_x < 0$ y $g_y < 0$

$$\cdot \text{ Si } C_x - \frac{g_x}{2} > 0 \rightarrow \delta > 0$$

$$\cdot \text{ Si } C_x - \frac{g_x}{2} < 0 \rightarrow \delta < 0$$

· Si el vector goal coincide con el eje X: $g_y = 0$

$$\cdot \text{ Si } g_x > 0 \text{ y } C_y > 0 \rightarrow \delta < 0$$

$$\cdot \text{ Si } g_x > 0 \text{ y } C_y < 0 \rightarrow \delta > 0$$

$$\cdot \text{ Si } g_x < 0 \text{ y } C_y > 0 \rightarrow \delta < 0$$

$$\cdot \text{ Si } g_x < 0 \text{ y } C_y < 0 \rightarrow \delta > 0$$

· Si el vector goal coincide con el eje Y: $g_x = 0$

$$\cdot \text{ Si } g_y > 0 \text{ y } C_x > 0 \rightarrow \delta < 0$$

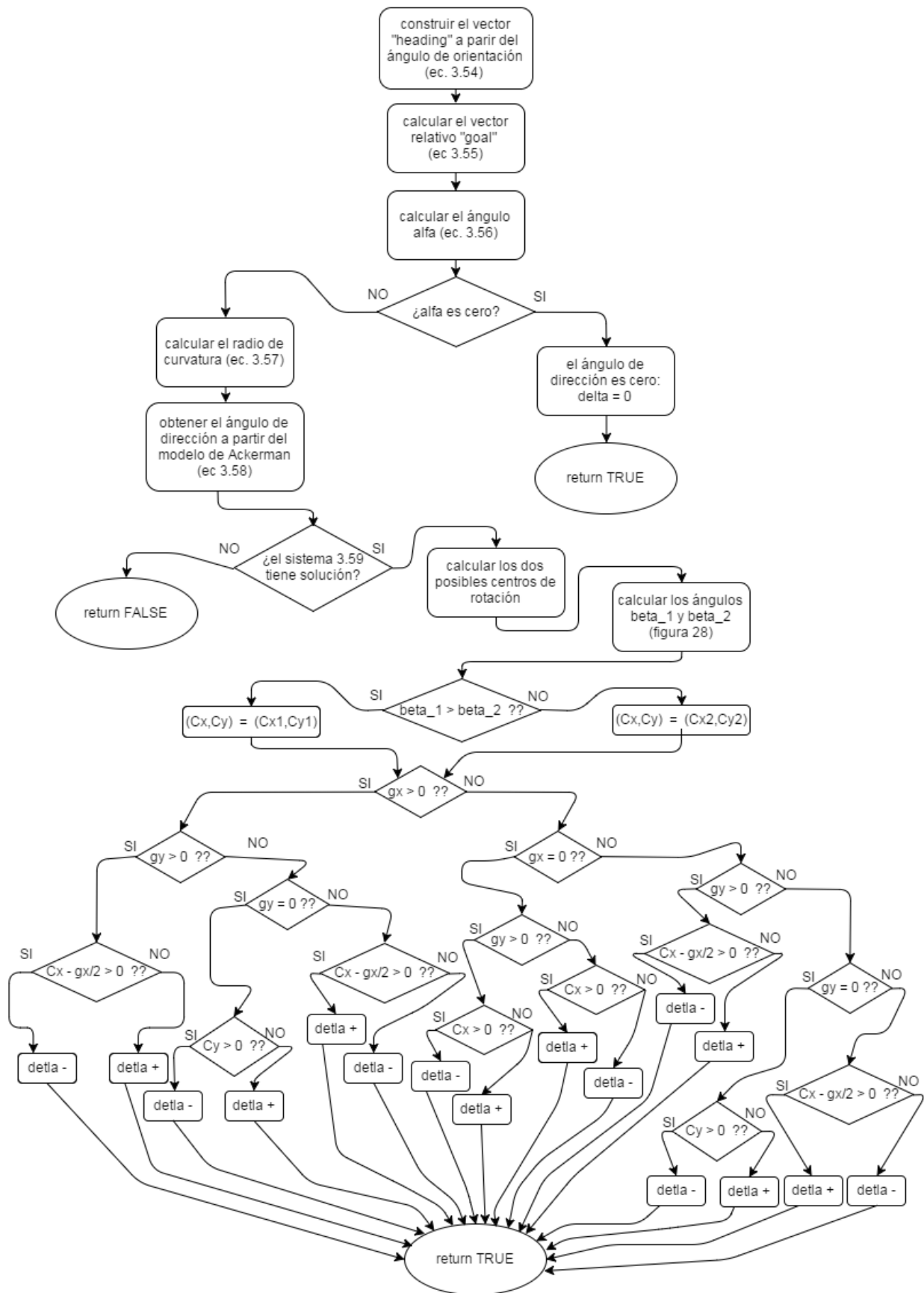
$$\cdot \text{ Si } g_y > 0 \text{ y } C_x < 0 \rightarrow \delta > 0$$

$$\cdot \text{ Si } g_y < 0 \text{ y } C_x > 0 \rightarrow \delta > 0$$

$$\cdot \text{ Si } g_y < 0 \text{ y } C_x < 0 \rightarrow \delta < 0$$

Resulta evidente, que éste criterio de signos es mucho más complejo que en el caso del método uno. La ventaja de este segundo método es que no se consideran la rotación relativa entre el sistema de referencia global y el sistema de referencia del vehículo. Por lo que no es necesario aplicar matrices de rotación para expresar un vector en uno de los sistemas de referencia cuando esté definido en el otro. Y es posible hacer sumas vectoriales directamente con los vectores calculados en cualquiera de los dos sistemas.

A continuación se expone el fujograma de una función que calcule el ángulo de dirección mediante el método 2.



Fujograma 2: cálculo del ángulo de dirección, método 2

3.2.4 Actualización de la orientación

Cómo último paso, a fin de realizar simulaciones para verificar el comportamiento teórico del algoritmo, se implementó un método para calcular la orientación del vehículo al final de cada movimiento.

Se recalca que ésta parte no es necesaria para la implementación real. No obstante, si es necesario actualizar la orientación del vehículo, pero la información acerca de la orientación instantánea del vehículo debe recibirse a través de sensores, no ser calculada de manera teórica. Si se desea es posible calcular la orientación de manera teórica y compararla con la orientación medida por los sensores, para utilizar dicha diferencia como realimentación en una ley de control.

Para hacer las simulaciones cuyos resultados se presentan en el capítulo 4, se hace la suposición de que el vehículo mantiene su trayectoria hasta alcanzar cada punto objetivo, y se calcula la orientación del vehículo en dicho punto. Como se comentó anteriormente, ésta no es la filosofía del algoritmo pure pursuit, ya que el método de trabajo es asignar un nuevo punto objetivo y comandar el vehículo hacia él siempre antes de que se alcance el punto objetivo anterior.

La señal para recalcular el punto objetivo y el nuevo ángulo de dirección puede ser tanto síncrona como asíncrona respecto al reloj del sistema. Si se prefiere una señal síncrona, el criterio para establecer la frecuencia de la señal debe estar basado en la velocidad del vehículo y el valor elegido para el parámetro “look ahead distance”, de tal forma que se garantice que la señal siempre se active antes de alcanzar el punto objetivo. Si se opta por una señal asíncrona, el criterio más apropiado se basa en activar la señal una vez que se ha recorrido una cierta distancia, por ejemplo un metro, aunque éste valor depende de la distancia que haya entre el vehículo y el punto objetivo. La implementación asíncrona requiere de un sistema que contabilice la distancia recorrida, lo cual se puede conseguir acoplado un odómetro en cualquiera de las cuatro ruedas.

A continuación se explica la obtención del nuevo ángulo de orientación del vehículo, el cual es utilizado por el algoritmo como orientación inicial para calcular el siguiente movimiento. Por claridad en la redacción se utilizará el nombre “heading” para referirse a la orientación del vehículo, o lo que es lo mismo, a la posición relativa del sistema de referencia del vehículo respecto al sistema de referencia global.

Cuando llega el momento de actualizar el “heading” del vehículo, se conocen todos los datos relevantes al movimiento, incluido las coordenadas del centro de giro. El problema geométrico a resolver es calcular el ángulo que forman dos vectores: el eje Y del sistema de referencia global, y el eje X del sistema de referencia del vehículo, cuando éste llega al punto objetivo. Las ecuaciones utilizadas deben por tanto estar referidas al sistema de referencia global. En la Figura 30 se representa gráficamente el problema a resolver.

ecuación matricial 3.38, lo cual implica calcular la matriz inversa de rotación. Se recuerda que la matriz de rotación de la expresión 3.38 es:

$$R(\beta) = \begin{pmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{pmatrix} \quad [3.58]$$

Donde β es el ángulo definido en la ecuación 3.39. La matriz 3.58 es ortogonal, por lo que el proceso de calcular su inversa es muy simple, ya que la inversa de una matriz ortogonal coincide con su traspuesta:

$$R^{-1}(\beta) = R^T(\beta) \quad [3.59]$$

No obstante, se demuestra a continuación que la ecuación 3.59 es correcta, para lo cual se supone que la matriz $R(\beta)$ no es ortogonal. De manera general, la matriz inversa de una matriz dada se obtiene según la expresión:

$$A^{-1} = \frac{1}{|A|} \cdot (A^*)^T \quad [3.60]$$

Aplicando la expresión 3.60 sobre la matriz de rotación 3.58 se tiene:

$$|A| \rightarrow |R(\beta)| = \cos^2 \beta + \sin^2 \beta = 1$$

$$A^* \rightarrow R(\beta)^* = \begin{pmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{pmatrix}$$

$$(A^*)^T \rightarrow (R(\beta)^*)^T = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix}$$

Por lo que finalmente, aplicando 3.60:

$$R^{-1}(\beta) = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \quad [3.61]$$

Por tanto, la regla para convertir un vector expresado en el sistema de referencia del vehículo, en el mismo vector expresado en el sistema de coordenadas global, es:

$$\begin{pmatrix} w_X \\ w_Y \end{pmatrix} = \begin{pmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{pmatrix} \begin{pmatrix} v_X \\ v_Y \end{pmatrix} \quad [3.62]$$

Aplicando la ecuación 3.62 con el vector \bar{c}_v como (v_X, v_Y) , se obtienen las coordenadas del vector respecto al sistema de referencia global. Y haciendo la siguiente suma de vectores, se obtiene finalmente el vector \bar{c}_w de la Figura 30:

$$c_w = c_v^* + V \quad [3.63]$$

Donde los vectores \bar{c}_w y \bar{V} son los definidos en la Figura 30, y el vector c_v^* es el resultado de aplicar 3.62 con $\bar{c}_v = (v_X, v_Y)$.

Con el vector \bar{c}_w calculado, se obtiene el vector \bar{h}_\perp haciendo la suma vectorial:

$$\bar{h}_\perp = \bar{g}_w - \bar{c}_w \quad [3.64]$$

Finalmente se calculan los dos posibles vectores perpendiculares a \overline{h}_\perp para lo cual se realiza la siguiente operación:

Siendo $\overline{h}_\perp = (h_{\perp x}, h_{\perp y})$:

$$\overline{h}_1 = (-h_{\perp y}, h_{\perp x})$$

$$\overline{h}_2 = (h_{\perp y}, -h_{\perp x})$$

Ambos vectores se representan en la Figura 31. Figura que también sirve para explicar el criterio utilizado para seleccionar el vector correcto.

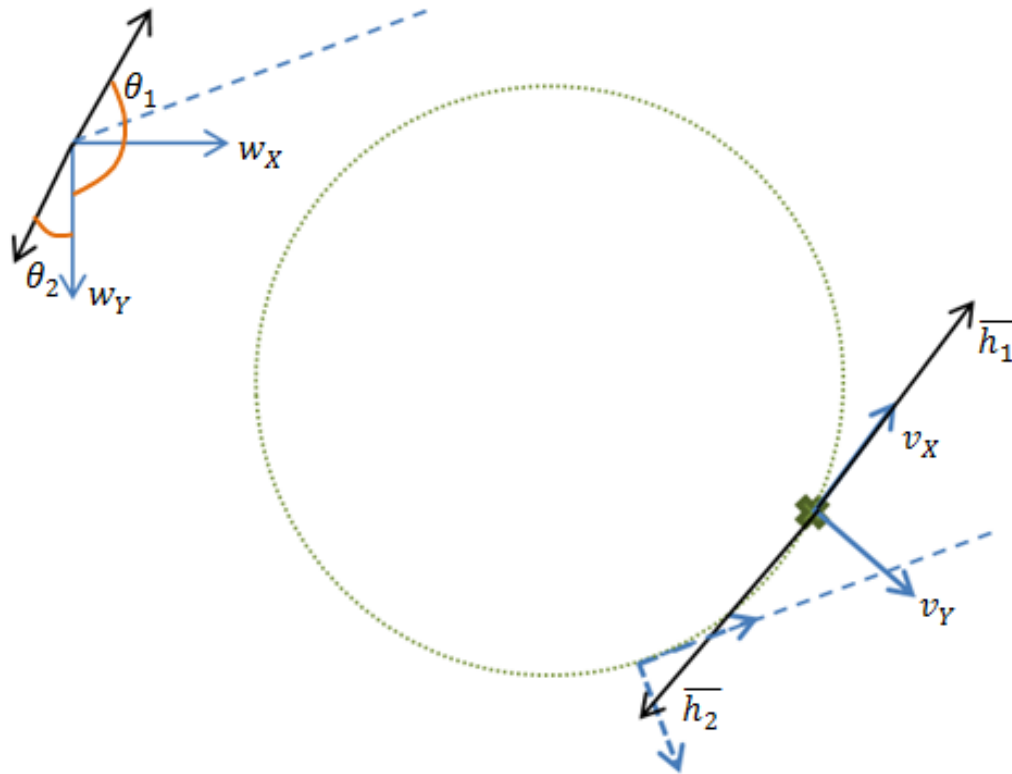


Figura 31: cálculo del nuevo heading y criterio

Para seleccionar el vector correcto se calculan los ángulos θ_1 y θ_2 según las expresiones:

$$\theta_1 = \cos^{-1} \left(\frac{h_{\perp x}}{\sqrt{h_{\perp x}^2 + h_{\perp y}^2}} \right) \quad [3.65]$$

$$\theta_2 = \cos^{-1} \left(\frac{-h_{\perp x}}{\sqrt{h_{\perp x}^2 + h_{\perp y}^2}} \right) \quad [3.66]$$

Los ángulos θ_1 y θ_2 son los posibles “heading” actualizados del vehículo, para saber cuál de ellos es el correcto, se comparan con respecto al heading inicial, denotado en la Figura 31 con la línea discontinua azul. Para ello se calculan los ángulos β_1 y β_2 definidos en la Figura 32.

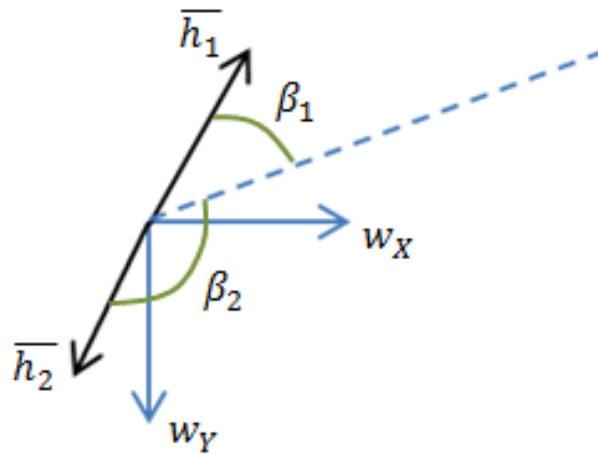


Figura 32: ángulos Beta_1 y Beta_2

Se establece el siguiente criterio:

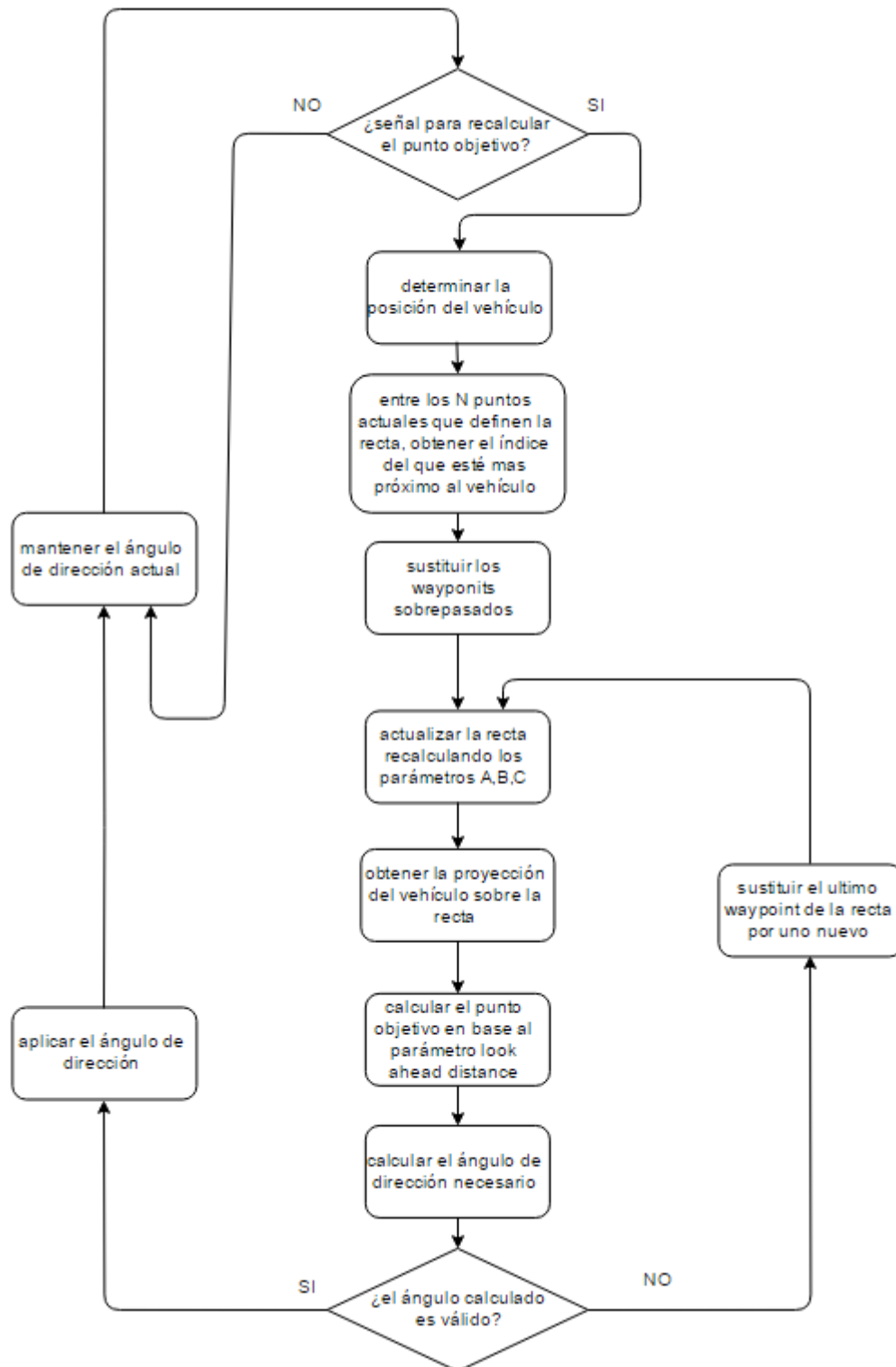
si $\beta_1 < \beta_2 \rightarrow \theta_1$ es el nuevo heading

si $\beta_2 < \beta_1 \rightarrow \theta_2$ es el nuevo heading

Con lo que finalmente, el “heading”, o ángulo de orientación del vehículo, queda actualizado, y ya es posible calcular el siguiente movimiento.

3.3 Diagrama de flujo del algoritmo completo

A continuación se presenta un flujograma que describe la ejecución del algoritmo completo. Los Flujogramas 1 y 2 amplían la información presentada aquí para el cálculo del ángulo de dirección.



Fujograma 3: algoritmo completo

IV

Resultados y Conclusiones

4.1 Resultados de las simulaciones

A fin de optimizar la trayectoria del vehículo, se realizaron algunas simulaciones con tres tipos de camino, los cuales se presentan a continuación. En todos los casos se modificaron dos variables:

- La generación de la trayectoria estableciendo distintos criterios para la elección de los waypoints con los que se define la recta.
- El valor del parámetro “look ahead distance”.

En todas las simulaciones se representan en azul los waypoints, en naranja los puntos objetivo calculados por el algoritmo, y en color gris la trayectoria teórica del vehículo. No obstante, cabe destacar que las simulaciones se han obtenido considerando que el vehículo se comanda hasta alcanzar el punto objetivo, lo cual como se ha comentado varias veces a lo largo de este informe, no es lo que sucede en la implementación real del algoritmo. La razón de haber hecho las simulaciones de esta manera es la dificultad que presenta estimar de manera teórica la posición instantánea del vehículo a lo largo de la curva que describe, ya que habría que recurrir al cálculo integral. Por otra parte, el único propósito de la simulación es hacerse una idea de la trayectoria que describe el vehículo, y el método implementado cumple con dicho objetivo.

Los caminos de ejemplo utilizados en las simulaciones se presentan en las figuras 31, 32 y 33.

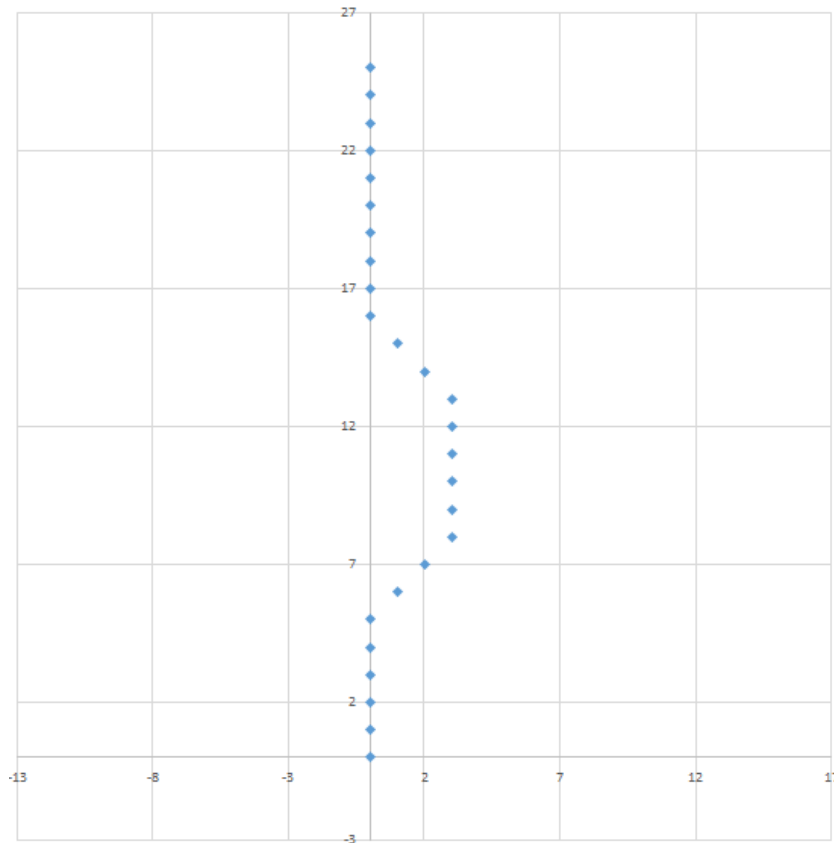


Figura 33: camino de ejemplo 1

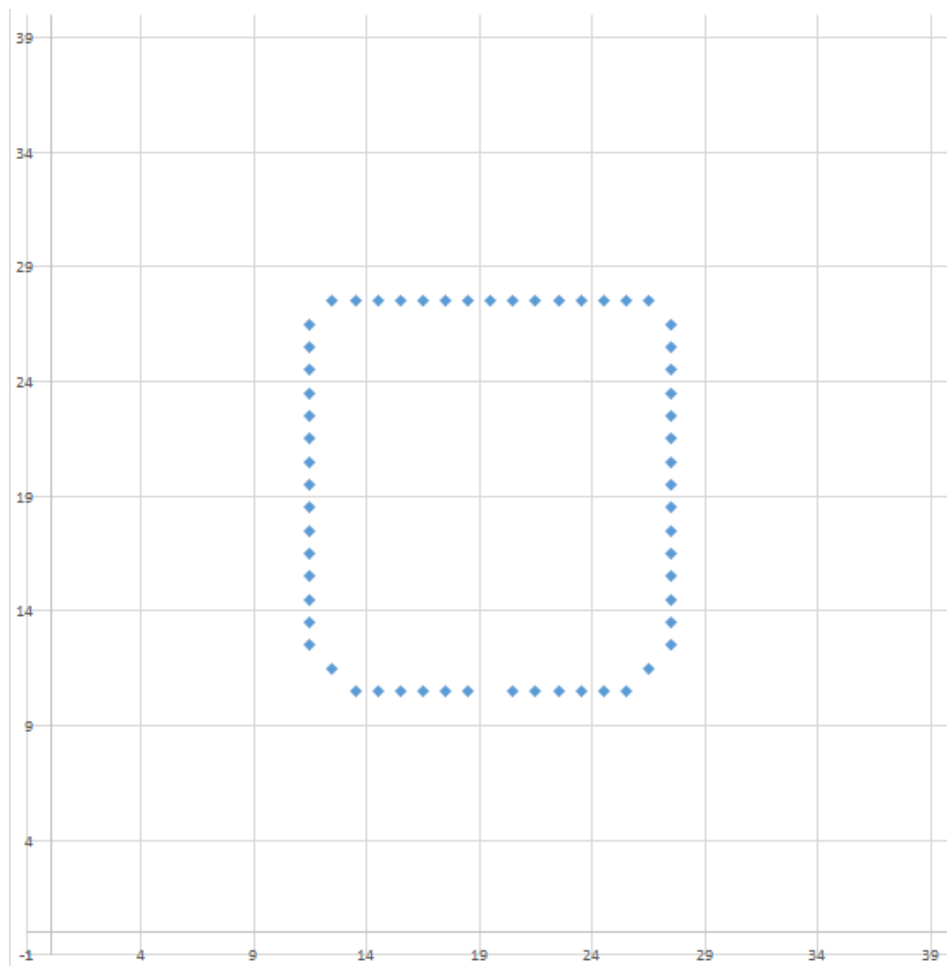


Figura 34: camino de ejemplo 2

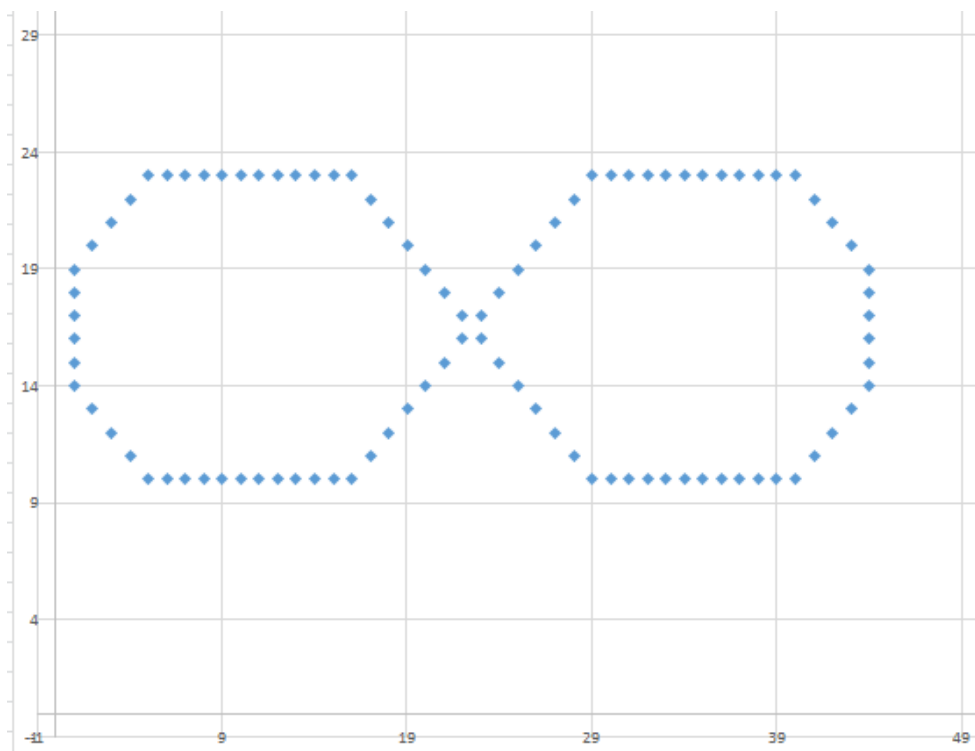
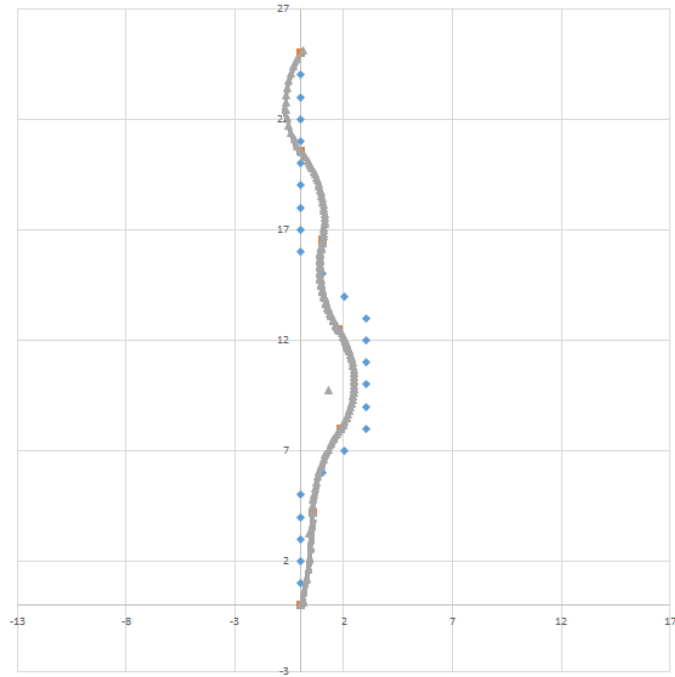


Figura 35: camino de ejemplo 3

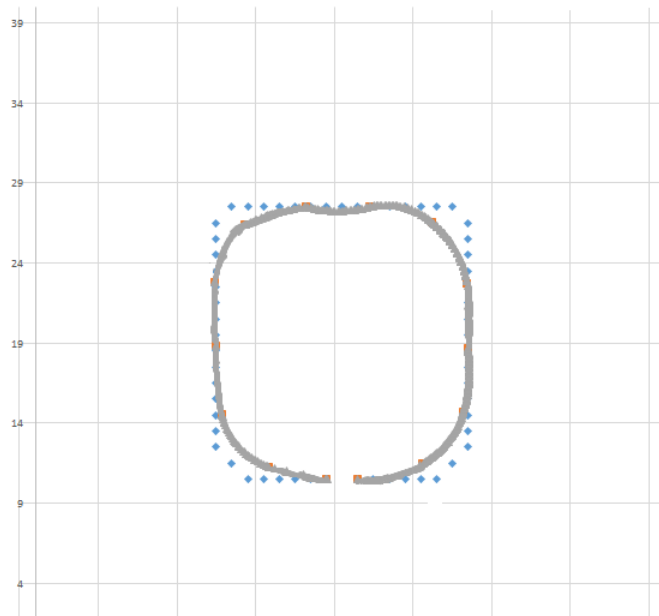
4.1.1 Criterio de selección de waypoints

A fin de establecer un criterio sobre el número de waypoints a seleccionar para definir la recta a la que aproximarse, sobre la cual se calcula el punto objetivo, se hicieron varias simulaciones con los tres caminos de ejemplo definidos anteriormente. El parámetro “look ahead distance” (LAD) se mantuvo constante para todas las simulaciones siendo 4 metros el valor elegido.

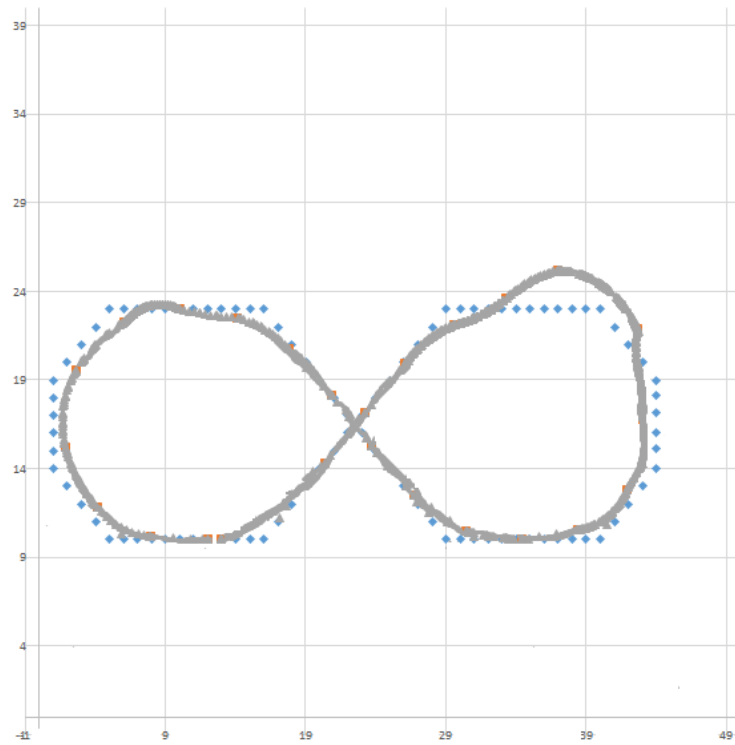
Caso de análisis 1: 10 waypoints en la dirección de avance del vehículo



Simulación 1: 10 WP, 4m LAD, camino 1

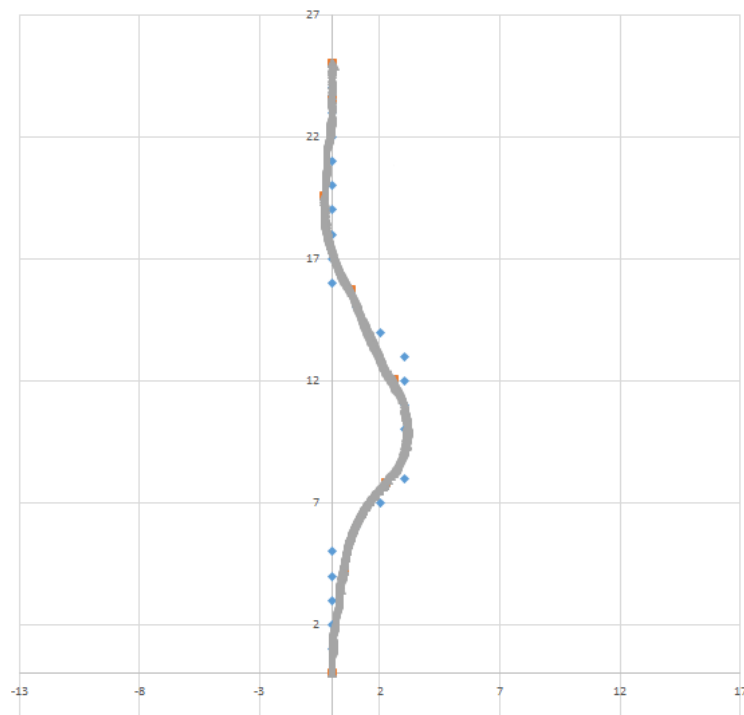


Simulación 2: 10 WP, 4m LAD, camino 2

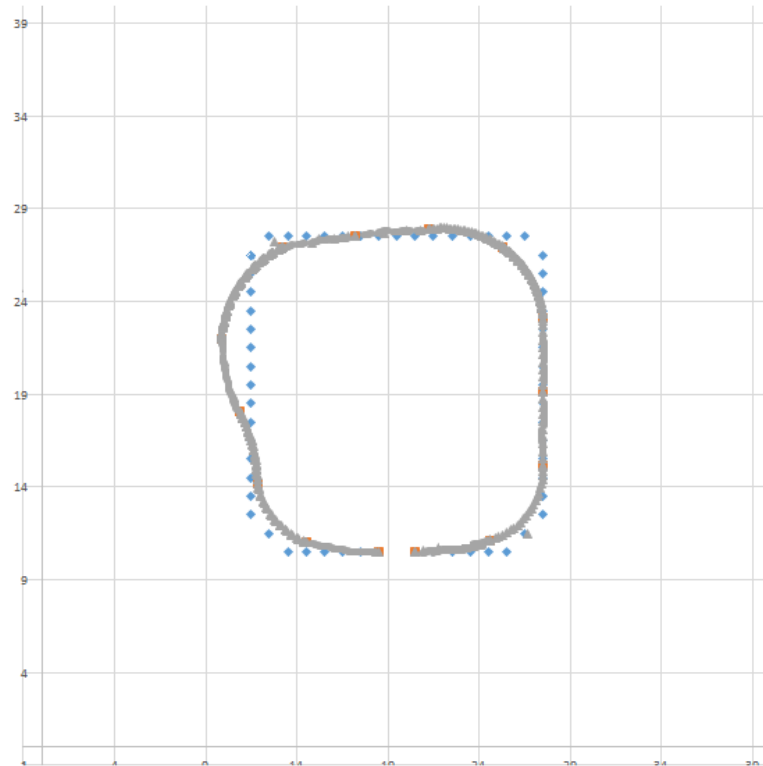


Simulación 3: 10 WP, 4m LAD, camino 3

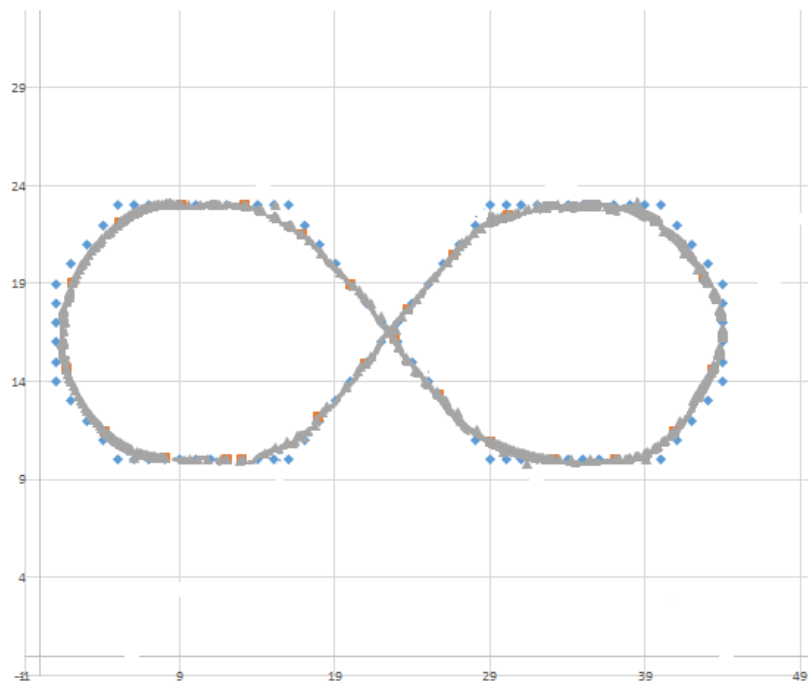
Caso de análisis 2: 8 waypoints en la dirección de avance del vehículo



Simulación 4: 8 WP, 4m LAD, camino 1

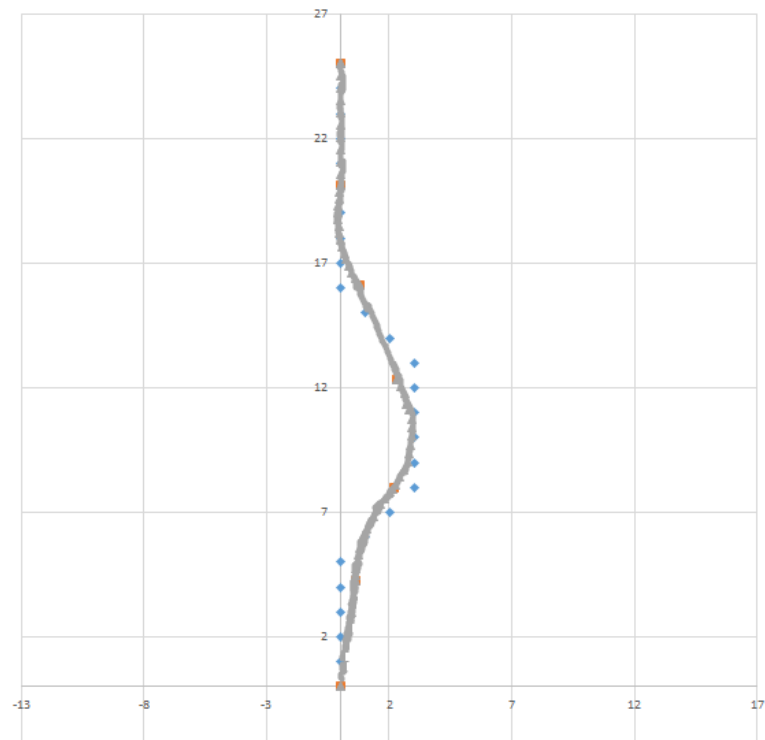


Simulación 5: 8 WP, 4m LAD, camino 2

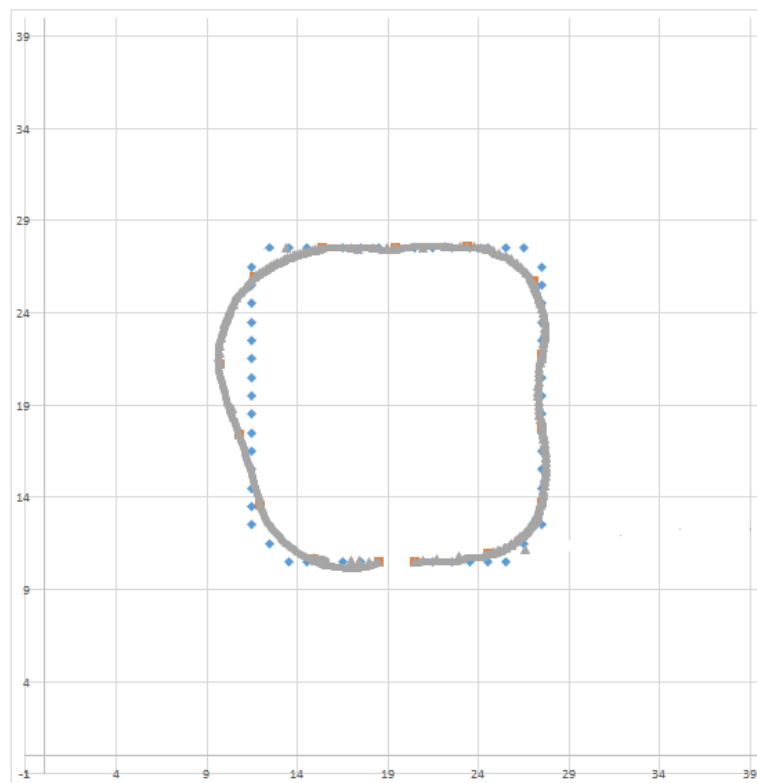


Simulación 6: 8 WP, 4m LAD, camino 3

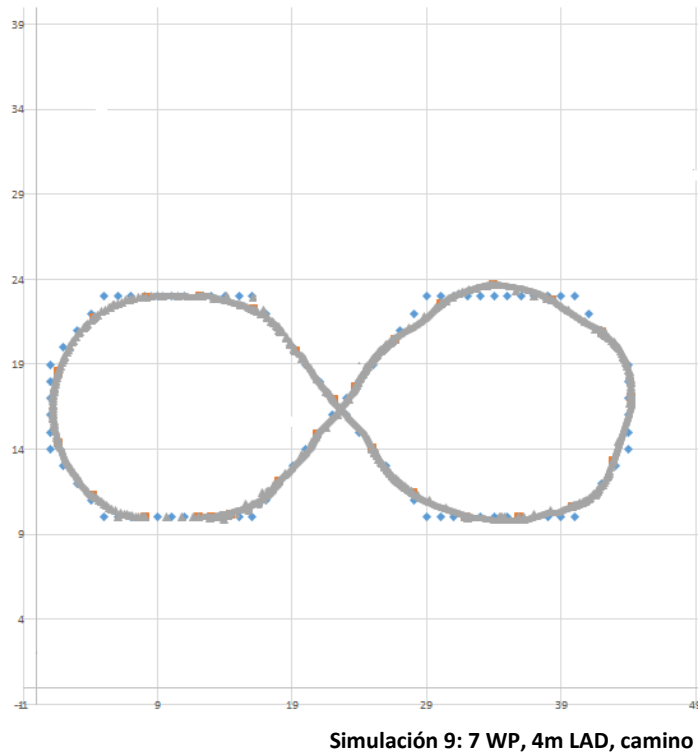
Caso de análisis 3: 7 waypoints en la dirección de avance del vehículo



Simulación 7: 7WP, 4m LAD, camino 1



Simulación 8: 7WP, 4m LAD, camino 2



4.1.2 Conclusiones sobre las simulaciones (número de waypoints)

Comparando los resultados de las simulaciones realizadas, se aprecia que se han obtenido los mejores resultados, en el conjunto de los tres caminos, con la configuración del “caso de análisis 2”. Es decir, eligiendo en total ocho waypoints en la dirección de avance del vehículo para construir la trayectoria a la que aproximarse.

Utilizando el camino de ejemplo uno (Figura 33), los resultados obtenidos con 7 y 8 waypoints son prácticamente iguales. En el caso de los 10 waypoints, se aprecia que la trayectoria teórica no se ajusta correctamente al camino deseado, el slalom que marcan los waypoints es demasiado pronunciado y corto como para que quede reflejada esa información al actualizar los puntos y volver a calcular la recta de aproximación, como resultado el vehículo zigzaguea.

El camino de ejemplo 2 (Figura 34) representa el rodeo de un obstáculo grande como una plaza o una fuente. Los resultados con 7 y 8 waypoints son una vez más similares y aceptables, con 10 waypoints la trayectoria se ajusta más al camino descrito por los waypoints pero el error de posición en los bordes es mayor que en los otros dos casos de análisis.

El camino de ejemplo 3 (Figura 35) representa un movimiento continuo, y en este caso se aprecia claramente que con 8 waypoints se ha conseguido el mejor resultado.

Por todo ello, finalmente el número de waypoints utilizados para construir la recta de aproximación se establece en 8. No obstante se recuerda que los resultados son teóricos y no experimentales, por lo que su utilidad se limita únicamente a hacer una primera aproximación.

4.1.3 Criterio para establecer el parámetro “look ahead distance”

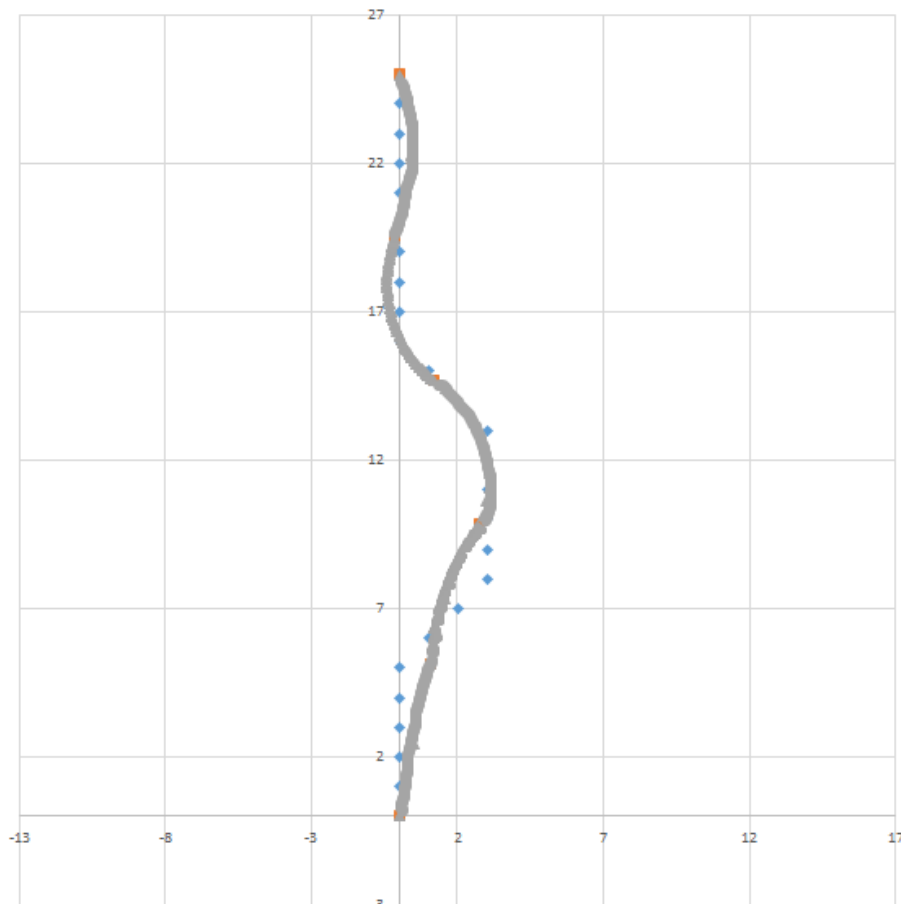
Como ya se ha comentado, la correcta elección del parámetro “look ahead distance” es de crucial importancia para el buen funcionamiento del algoritmo de navegación. Los efectos derivados de su elección se presentaron en la sección 2.2.1. En la práctica este parámetro se ajusta experimentalmente. No obstante, se realizaron algunas simulaciones teóricas para hacer una primera aproximación.

A fin de establecer un criterio para elegir el parámetro “look ahead distance” se realizaron simulaciones teóricas con los caminos de ejemplo definidos al principio de ésta sección, eligiendo valores de 4, 5 y 6 metros. Se utilizaron en todos los casos, ocho waypoints para definir las rectas de aproximación.

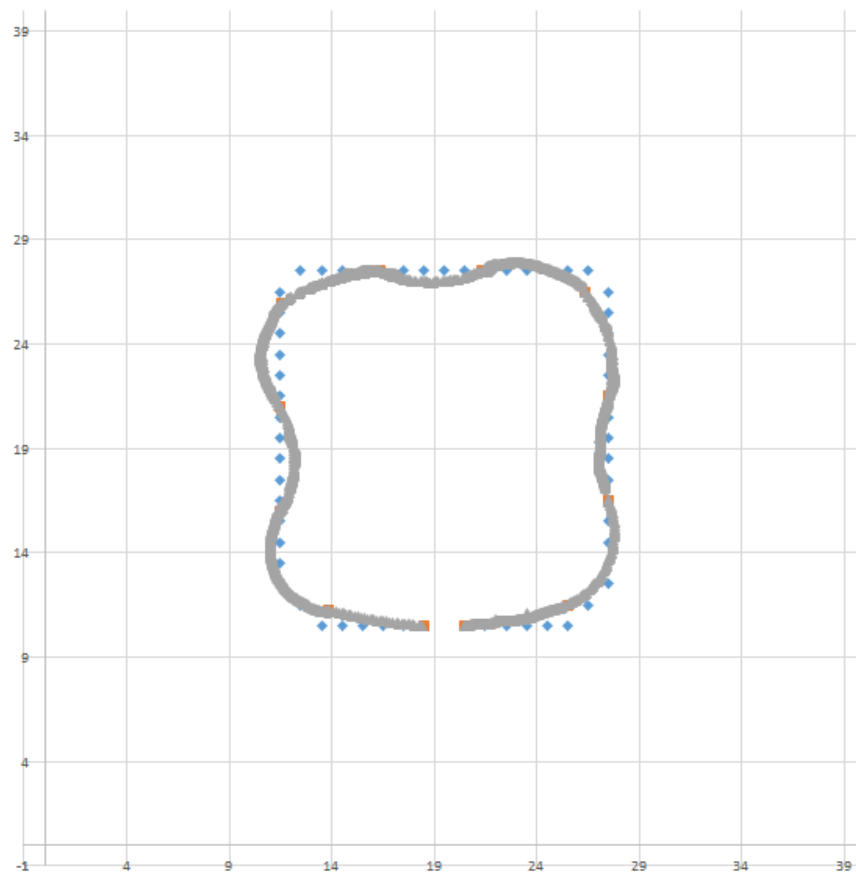
Caso de análisis 1: 4m LAD (look ahead distance)

Las simulaciones corresponden con el caso de análisis 2 de la sección 4.1.1, (Simulación 4, Simulación 5, Simulación 6).

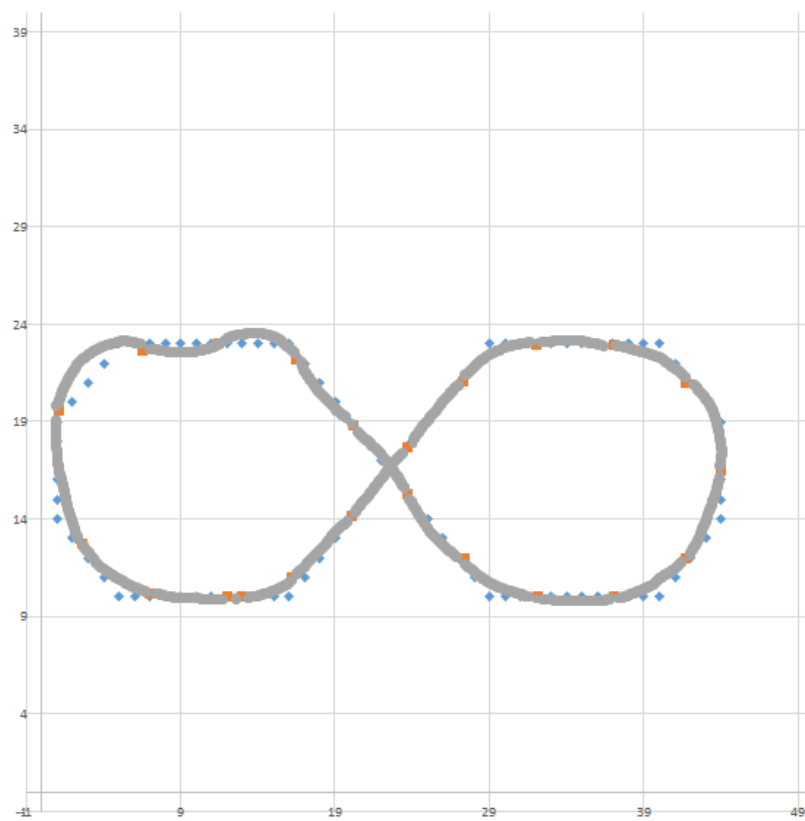
Caso de análisis 2: 5m LAD (look ahead distance)



Simulación 10: 8WP, 5m LAD, camino 1

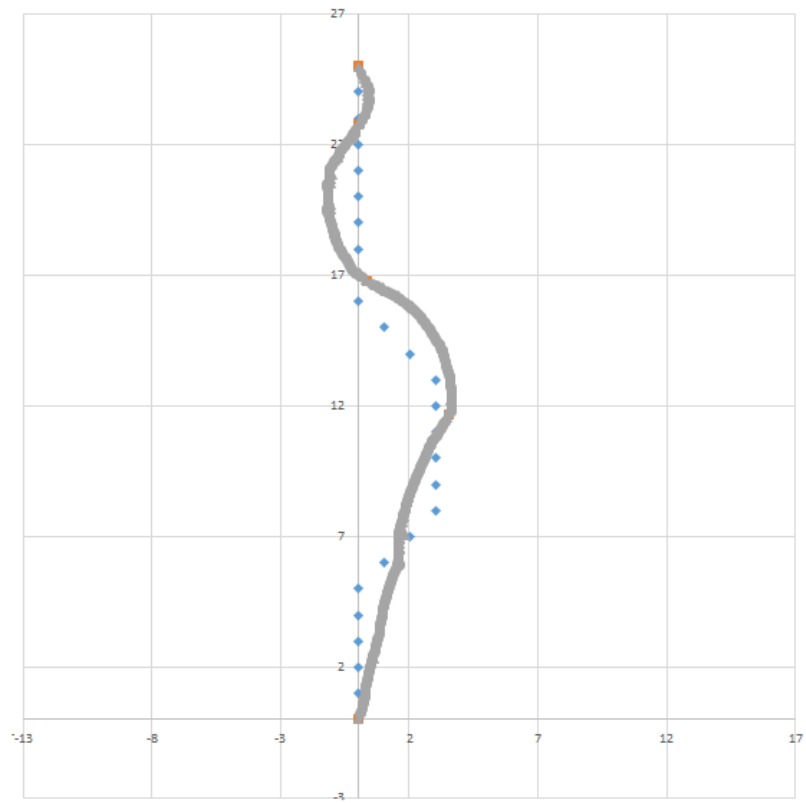


Simulación 11: 8WP, 5m LAD, camino 2

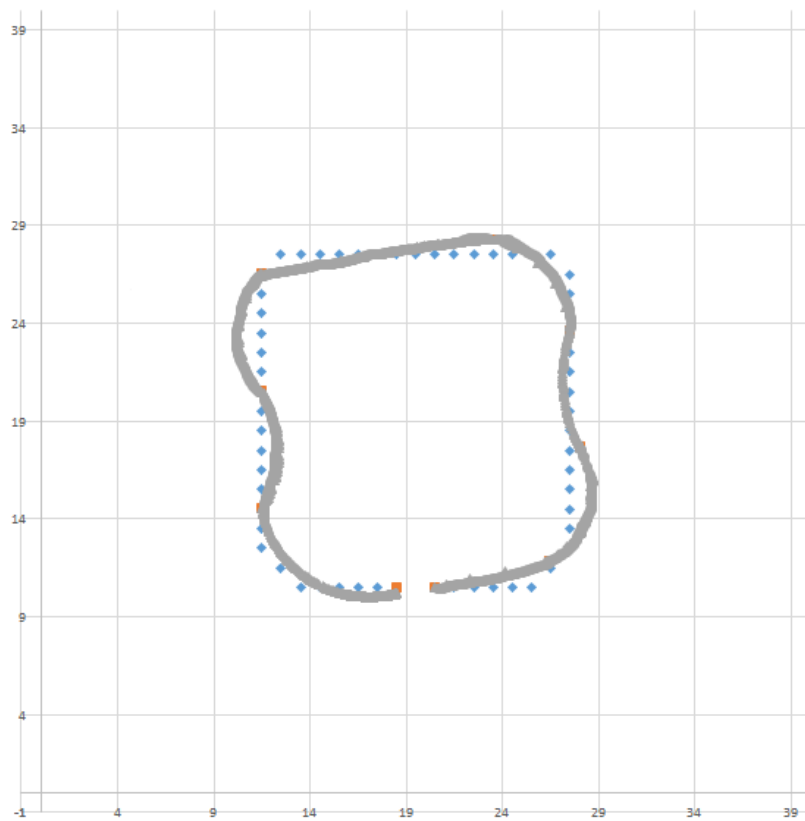


Simulación 12: 8WP, 5m LAD, camino 3

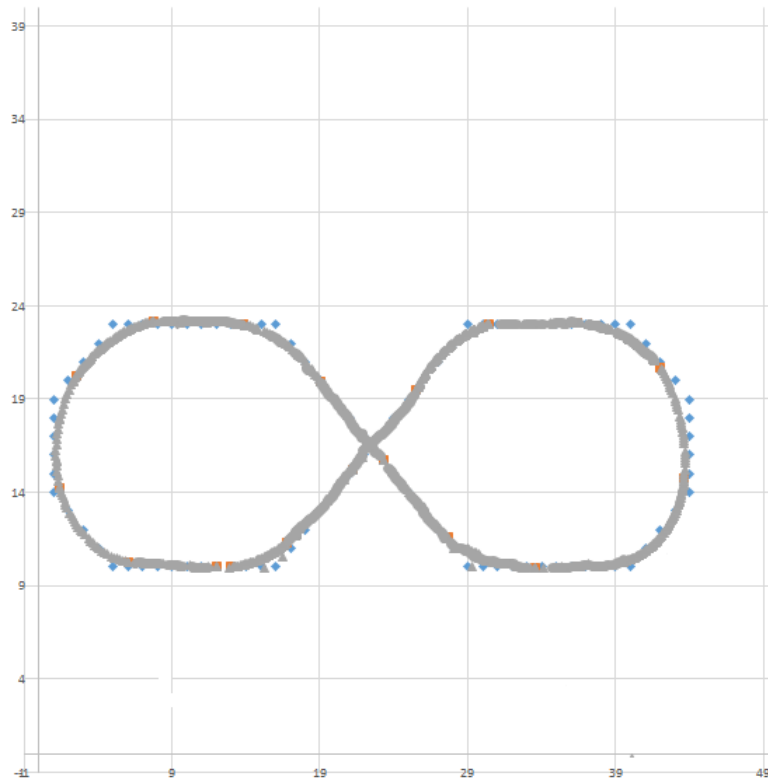
Caso de análisis 3: 6m LAD (look ahead distance)



Simulación 13: 8 WP, 6m LAD, camino 1



Simulación 14: 8 WP, 6m LAD, camino 2



Simulación 15: 8 WP, 6m LAD, camino 3

4.1.4 Conclusiones sobre las simulaciones (*look ahead distance*)

En primer lugar, recalcar que las trayectorias teóricas representadas en las simulaciones no corresponden con la ejecución real del algoritmo, ya que se está suponiendo que el vehículo alcanza en cada movimiento el punto objetivo calculado, cuando en la realidad eso nunca debe suceder.

Obviando este hecho, vemos que los mejores resultados se obtienen al utilizar un valor de 4 ó 5 metros. Dada la distribución de los waypoints y la separación entre ellos, un valor de entre 4 y 5 metros hace que el punto objetivo calculado esté aproximadamente en el medio de entre los 8 waypoints que se utilizan para definir la recta de aproximación.

Los resultados teóricos para el camino 1, son aproximadamente iguales en el caso de los 4 metros (Simulación 4) y el de los 5 metros (Simulación 10). Sin embargo, comparando los resultados en los caminos 2 y 3 para estos mismos valores de “look ahead distance”, el algoritmo da mejores resultados utilizando los 4 metros (Simulación 5, Simulación 6).

Emplear valores mayores que éstos, como 6 metros por ejemplo, provoca que el vehículo se desvíe más del camino deseado, por lo que inicialmente se descartan valores superiores a 5 metros para la implementación real.

Por todo ello, el algoritmo se ha desarrollado utilizando 4 metros como el valor del parámetro “look ahead distance”. No obstante, este es un parámetro que debe ajustarse de manera empírica una vez se hace la implementación real.

4.2 Conclusiones y Líneas futuras

A la luz de los resultados obtenidos, se considera que se ha cumplido el objetivo marcado al inicio de este proyecto. Se ha logrado transformar la información procedente de los waypoints en directrices que permitan comandar el vehículo a través del camino deseado.

No obstante, al término de este proyecto, el algoritmo no se ha probado en el vehículo real, por lo que no es posible afirmar aún su funcionalidad. Para poder realizar pruebas con el vehículo, es necesario en primer lugar adaptar el código desarrollado para poder ejecutarlo sobre el sistema operativo ROS, e integrarlo con el resto de procesos que conforman el sistema de control del vehículo.

Una vez adaptado el código será necesario optimizar el valor de los parámetros “look ahead distance” así como del número de waypoints que se utilicen para determinar la trayectoria hacia la que comandar el vehículo.

Los resultados obtenidos en las simulaciones presentadas en el apartado 4.1 no pueden considerarse concluyentes. Como se dijo, el propósito de la simulación era obtener una primera aproximación de la trayectoria real del vehículo. En los cálculos de la simulación se está suponiendo que el vehículo alcanza en cada movimiento el punto objetivo calculado, cuando en la realidad eso nunca debe suceder.

Los sensores instalados actualmente en el vehículo permiten conocer la distancia que éste ha recorrido, por tanto se considera que la mejor solución para implementar el algoritmo pure pursuit en este caso, es recalcular el punto objetivo cada vez que el vehículo recorra una distancia determinada. Por supuesto dicha distancia debe ser menor que el valor del parámetro “look ahead distance”, para que el vehículo nunca alcance el punto objetivo al que se dirige, siguiendo así la filosofía del método Pure Pursuit, cuya traducción al español sería “persecución continua”.

Si por ejemplo se utilizan 4 metros como valor del parámetro “look ahead distance”, y se establece que el punto objetivo sea recalculado cada 70 cm recorridos por el vehículo, la frecuencia aproximada de la señal para recalcular el punto objetivo sería de unos 7 Hz, considerando una velocidad media del vehículo de 18Km/h.

La generación de la trayectoria a la que aproximarse es otra línea sobre la que se puede trabajar a fin de optimizar el resultado final algoritmo. En este proyecto se ha optado por ajustar la nube de puntos a una función lineal, no obstante es posible que se obtengan mejores resultados utilizando otro método de ajuste de curvas, o mejorando el actual. Por ejemplo, durante las simulaciones se hicieron ensayos con una distancia entre waypoints de 0,5 metros en lugar de un metro. De esta manera se consigue una mayor precisión en la definición del camino a seguir, y en consecuencia la variación de la pendiente de la recta es más suave al cambiar los puntos que la definen.

V

Planificación y presupuesto

5.1 Planificación

El objetivo de este trabajo fin de grado era desarrollar un algoritmo de navegación, que fuera capaz de convertir un “camino de puntos” dado (waypoints) en instrucciones con las que el vehículo pudiera navegar de manera autónoma.

Para lograr el objetivo descrito a largo plazo, se estableció un plan de trabajo dividido en tareas individuales con objetivos alcanzables a corto plazo. De esta manera se dividió el proyecto en 3 fases:

Fase 1: Planificación. (4 semanas)

Las primeras cuatro semanas se dedicaron a la búsqueda y recopilación de información sobre métodos de seguimiento de trayectorias (“path tracking”) y su implementación. La investigación reveló que existen múltiples algoritmos de “path tracking”, que pueden agruparse según los principios en que están basados para lograr el seguimiento de la trayectoria, a saber: métodos geométricos, métodos basados en un modelo cinemático del vehículo, y métodos basados en un modelo dinámico del vehículo. De todos ellos los más populares son los basados en la geometría, por su mayor simplicidad y sus buenos resultados.

Fase 2: Definición (2 semanas)

Después de recopilar información acerca de los algoritmos de generación de trayectorias existentes, se decidió cuál de ellos implementar en base a las características y necesidades de la aplicación. El algoritmo elegido fue el “Pure Pursuit” por los motivos que se explicaron en la sección 2.2.4

Para comprobar la viabilidad de la solución, antes de empezar a desarrollar el código completo se dedicaron otras dos semanas a elaborar una hoja Excel, con la finalidad de obtener una simulación y un pronóstico del resultado final.

Fase 3: Desarrollo y ejecución (18 semanas)

Los siguientes cuatro meses se dedicaron al desarrollo y optimización del código, además de a la depuración de errores, hasta alcanzar una versión válida. Posteriormente se realizaron algunas simulaciones para visualizar la posible trayectoria teórica del vehículo.

En su conjunto, el desarrollo del presente proyecto ha durado aproximadamente un semestre, incluyendo la redacción de este documento.

5.2 Presupuesto

Para el desarrollo de este proyecto no ha sido necesaria la adquisición de ningún componente físico, puesto que se disponían de los medios necesarios y a condición de ser alumno de la universidad, las licencias de software utilizadas son accesibles a coste cero.

No obstante, para llevar a cabo el desarrollo del código, han sido necesarios los siguientes medios, cuyo valor se estima en:

- Ordenador de sobremesa (600€)
- Entorno de programación, licencia de Microsoft Visual Studio (646€)
- Paquete Microsoft Office (149€)

Se considera que el año tiene 360 días y 1700 horas laborales. Estableciendo una amortización de 5 años para todos los productos, tanto hardware como software, se obtiene:

$$Amortización_{PC} = \frac{600 \cdot 360}{1700 \cdot 5} = 25,41 \text{ €}$$

$$Amortización_{Visual\ Studio} = \frac{646 \cdot 360}{1700 \cdot 5} = 27,36 \text{ €}$$

$$Amortización_{office} = \frac{149 \cdot 360}{1700 \cdot 5} = 6,31 \text{ €}$$

Incluyendo los costes en recursos humanos, y considerando que 1 crédito ECTS corresponde a 25 horas de trabajo, la duración total del desarrollo del proyecto asciende a 300 horas. Se supone un sueldo de 15€/hora para un ingeniero junior, y de 25€/hora para un ingeniero sénior. Estimando que el ingeniero sénior dedica el 10% del tiempo total del desarrollo del proyecto en supervisar y orientar al ingeniero junior, se establecen los siguientes costes en materia de recursos humanos:

$$Recursos\ humanos = 300 \cdot 15 + 300 \cdot 0.1 \cdot 25 = 5.250\text{€}$$

Sumando todos los conceptos, se tiene un presupuesto total de:

$$presupuesto\ total = 25,41 + 27,36 + 6,31 + 5.250 = 5.309,08 \text{ €}$$

Aplicando el 21% de la suma en concepto de impuestos, se obtiene que el presupuesto final del proyecto asciende a:

$$presupuesto\ final = 5.309,08 \cdot 1.21 = 6.423,99\text{€}$$

APÉNDICE

Fuentes Bibliográficas

- [1] "Fundamentals of Vehicle Dynamics". Thomas D. Gillespie. SAE (1992)
- [2] "Automatic Steering Methods for Autonomous Auto-mobile Path Tracking". Jarrod M. SNIDER. Robotics Institute & Carnegie Mellon University. Pittsburgh, Pennsylvania 2009
- [3] "Introduction to Autonomous Mobile Robots". Roland SIEGWART, Illah R. NOURBAKHSH. Massachusetts Institute of Technology & 2004
- [4] "Path Tracking for a Miniature Robot" Martin Lundgren. Department of Computing Science Umeå University Sweden, 2003
- [5] "Mathworks documentation" <http://es.mathworks.com/help/robotics/ug/pure-pursuit-controller.html>
- [6] "Vector Pursuit path tracking for Autonomous Ground Vehicles". Jeffery S. Wit. PhD thesis, University of Florida, 2000.
- [7] "A geometrical introduction to screw theory" E. Minguzzi Dipartimento di Matematica Applicata "G. Sansone", Università degli Studi di Firenze, Via S. Marta 3, I-50139 Firenze, Italy
- [8] "Technician's repair and service manual" EZGO a TEXTRON company 2001. 28646-G01
- [9] R C. Coulter; Implementation of the Pure Pursuit Path Tracking Algorithm, Robotics Institute, Carnegie Mellon University, January, 1992
- [10] G. Dudek, M. Jenkin; Computational Principles of Mobile Robotics, Cambridge University Press, New York, 2000
- [11] página web del departamento de automática y sistemas de la universidad Carlos III de Madrid:http://portal.uc3m.es/portal/page/portal/dpto_ing_sistemas_automatica/investigacion/IntelligentSystemsLab/research/auv#iCab

Anexo 1

Base Part Number: 76180



MODEL: FREEDOM SE
TYPE: ELECTRIC POWERED, PERSONAL GOLF CAR
MODEL YEAR: 2006

PRODUCT SPECIFICATION

Motor:..... 36 volt DC series wound, brazed armature, solid copper windings. Non-vented 2.5 hp (1.9 kW) @ 2700 RPM (one hour)

Drive Train: Direct motor shaft connected to transaxle pinion shaft

Electrical System: 36 volt DC, six, 6 volt deep cycle storage batteries (115 minute minimum, 225 amp-hour @ 20 Hr. discharge rate)

Speed Control: Solid state continuously variable speed controller. 300 Amp capacity, inductive throttle sensor.

Charger: Fully automatic line compensating, 21 amp DC output at 36 volts; input 110 - 120 volts, 10.5 amps, 60 cycle AC, Underwriters Laboratories (U.L.) Listed, C.S.A. Certified. PowerWise™II portable

Transaxle: Differential with helical gears, 12.44:1 ratio

Brakes: Dual rear wheel mechanical self-adjusting drum brakes. Automatic park brake release with self-compensating system

Front Suspension: Leaf springs with hydraulic shock absorbers

Rear Suspension: Leaf springs with hydraulic shock absorbers

Steering: Self-compensating single reduction rack and pinion

Speed: 13 mph \pm 0.5 mph (21 kph \pm 0.8 kph) Level ground

Weight: 550 lb (250 kg), (without batteries)

Length: 92.6 in (235 cm)

Width: 46.5 in (118 cm)

Height: 48.6 in (123 cm) (at top of steering wheel)
67 in (170 cm) (top of sun canopy)

Ground Clearance: 4.3 in (11 cm) (at differential)

Wheel Base: 65.5 in (166 cm)

Front Wheel Tread: 33.5 in (85 cm)

Rear Wheel Tread: 38.0 in (97 cm)

Turniong Clearance Dia: 18.9 ft (5.72 m)

Load Capacity: 800 lb (360 kg) including operator, passenger, accessories and cargo.

Seating: Vacuum formed fabric backed vinyl covers over cushion foam. Standard seating for operator and one passenger

Body: Injection molded TPE (Thermoplastic Elastomer)

Color & Finish: Special Color Options. Base coat/clear coat finish

Chassis: Welded tubular steel with Durashield™ powder coat paint

Operator Features: Removable dash mounted key switch, reverse warning indicator, "deadman (operator presence switch)" accelerator control, integral handgrip on hip restraints, manual forward/reverse selector

Dash Panel: Tee, ball, and 4 position drink holders of glass fiber reinforced thermoplastic olefin

Steering Wheel: 13 in (33 cm) diameter with dual handgrips

OPTIONS/ACCESSORIES:

State of Charge Meter (LED) in dash	S	Safety:	
Front Tires:		Horn	S
Links 18 x 8.50 x 8 (4 Ply Rated)	S	Convenience:	
Sure Trail 18 x 8.50 x 8 (Load Range C)	O	2 Bag Attachment	O
Tour Max 18 x 8.00 x 10 mounted on Alloy Wheels	O	4 Bag Attachment	O
Rear Tires:		Ash Tray	
Links 18 x 8.50 x 8 (4 Ply Rated)	S	Bag Cover	O
Sure Trail 18 x 8.50 x 8 (Load Range C)	O	Bunker Rake with Holder	O
Turf Saver 18 x 8.50 x 8 (4 Ply Rated)	O	CE Package	O
Tour Max 18 x 8.00 x 10 mounted on Alloy Wheels	O	Center Sweater Basket	S
Cosmetic:		Cooler Jug (No Side Basket)	O
Non Standard Paint Color	O	Differential Scuff Guard	O
Non Standard Seat Color	O	Divot Repair (Sand Bottle, No Side Basket)	O
Pin Stripes	O	Divot Repair (Sand Container & Scoop (Requires Side Basket))	O
Special Color Options	S	Fender Scuff Guard	O
Special Seat Color Options	S	Flip-Flop Kit (Convertible Carrier)	O
Wheel Covers:		Fold down windshield. Requires Sun Canopy (Top)	O
6 in (15 cm) Chrome	O	Hour Meter	O
8 in (20 cm) Chrome	O	Ice Chest & Mounting Bracket	O
Spoke, Black & Gold	O	Individually Keyed Switch	O
Spoke, Black & Silver	O	Locking glove boxes	O
Wood Grain Dash	O	Message Holder	O
Lighting Package:		Rear Facing Seat	O
Brake Lights	S	Rear View Mirror. Requires Sun Canopy (Top)	O
Headlights	S	Side Basket	O
Taillights	S	Soft Swipe Spike Cleaner	O
Turn Signals with 4-Way Flashers	O	Sun Canopy (Top)	O
		Tow Bar (Casual)	O
		Tow Bar (Permanent)	O
		Weather Protection (3 or 4 Sided). Requires Sun Canopy (Top)	O
		Windshield. Requires Sun Canopy (Top)	O

S = Standard Equipment**O** = Option or Accessory